UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
11/592,786	11/03/2006	Amar Kumar	247-050US	2249

131475   7590   05/27/2016

Dilworth IP - SAP
2 Corporate Drive, Suite 206
Trumbull, CT 06611

EXAMINER
SCHALLHORN, TYLER J

ART UNIT	PAPER NUMBER
2177	

MAIL DATE	DELIVERY MODE
05/27/2016	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE
_____

BEFORE THE PATENT TRIAL AND APPEAL BOARD
_____

Ex parte AMAR KUMAR and WOLFGANG E. WALTER
_____

Appeal 2014-008976
Application 11/592,786
Technology Center 2100
_____

Before THU A. DANG, CARL L. SILVERMAN, and NORMAN H.
BEAMER, *Administrative Patent Judges*.

DANG, *Administrative Patent Judge*.

DECISION ON APPEAL

I. STATEMENT OF THE CASE

Appellants appeal under 35 U.S.C. § 134(a) from the Examiner's
Final Rejection of claims 1, 3–10, and 12–19. Claims 2 and 11 have been
canceled. We have jurisdiction under 35 U.S.C. § 6(b).

We affirm.

A. INVENTION

According to Appellants, the invention relates to "capturing user
interface information to recreate a state of the user interface at a later time or
different place" (Spec. ¶ 1).

B. ILLUSTRATIVE CLAIM

Claim 1 is exemplary:

1. A method comprising:

converting a screen layout into an Extensible Markup Language ("XML") string wherein converting includes identifying an internal state of a plurality of software functions each function of the plurality associated with a portion of the screen layout at least a plural subset of the software functions having associated therewith each at least one user input field, each user input field residing at a defined location within the screen layout and incorporating the internal state for each software function into the XML string and wherein the internal state includes the defined location and current content of the user input fields associated with the plurality of software functions; and

recreating the screen layout from the XML string at a different place or time, at least in part, by supplying the internal state information from the XML string to each software function to return the software functions to the internal state from the XML string and repopulating the user input fields residing at the defined locations from the software function based on that internal state.

C. REJECTION

The prior art relied upon by the Examiner in rejecting the claims on appeal is:

Dando	US 2003/0058286 A1	Mar. 27, 2003
Tafoya	US 6,829,607 B1	Dec. 7, 2004
Di Franco	US 2005/0125715 A1	June 9, 2005
Cutler	US 2005/0188329 A1	Aug. 25, 2005

David Powers, *Foundation PHP for Dreamweaver 8*, 17–21 (Chris Mills ed., Dec. 19, 2005) ("Powers").

Document Object Model Core, 1–51 (Arnaud Le Hors et al. eds., Nov. 13, 2000) ("Le Hors").

Claims 1, 3–8, 10, and 14–18 stand rejected under 35 U.S.C. § 103(a) as unpatentable over the teachings of Dando, Powers, and Di Franco.

Claim 5 stands rejected under 35 U.S.C. § 103(a) as unpatentable over Dando, Powers, Di Franco, and Le Hors.

Claim 9 stands rejected under 35 U.S.C. § 103(a) as unpatentable over Dando and Di Franco.

Claim 12 stands rejected under 35 U.S.C. § 103(a) as unpatentable over Dando, Di Franco, and Tafoya.

Claim 13 stands rejected under 35 U.S.C. § 103(a) as unpatentable over Dando, Di Franco, and Cutler.

Claim 19 stands rejected under 35 U.S.C. § 103(a) as unpatentable over Dando, Di Franco, Powers, and Official Notice.

II. ISSUE

The principal issue before us is whether the Examiner erred in finding the combination of Dando, Powers, and Di Franco teaches or would have suggested "converting" a screen layout into an XML string, which includes "identifying *an internal state* of a plurality of *software functions* each function of the plurality *associated with a portion of the screen layout* at least a plural subset of the software functions having associated therewith each at least one *user input field*," each user input field "residing at a defined location within the screen layout and incorporating the internal state for each software function into the XML string," wherein "the internal state *includes the defined location and current content of the user input fields* associated with the plurality of software functions" (claim 1) (emphases added).

3

III. FINDINGS OF FACT

The following Findings of Fact (FF) are shown by a preponderance of the evidence.

Di Franco

1. Di Franco discloses a method "for the storage of display values from one or more fields of a form of a graphical user interface (GUI) application running on a computing device" (Abstr.). A user "can fill in a form with data the user selects and then store that selected data in a data storage file and then when the user requires, that selected data can be used to repopulate a form opened by the user at any time" (¶ 15). The invention includes "software code which in response to a load command by a user in relation to an open form and a designated or default data storage file, populates the fields of the form with the data stored in the file" (¶ 11). Further, "a correlation for each field of the form may be stored in the data storage file as name/value pairs, where for each field of the form the name is an element identifier in the GUI application and the value is the display value for that field" (¶ 16). The user "has the ability to save display values from any form at any point in time" (¶ 85).

Dando

2. Dando discloses a "configurable user-interface component management system . . . where frame containers contain a number of child components and user-interface components" (Abstr.). "A view component displays the information of a particular application and provides an area where a user enters data and executes action on that data" (¶ 60).

IV. ANALYSIS

Appellants contend that in Di Franco, "there is no association between location on the screen and the content of a field" (App. Br. 10). According to Appellants, while Di Franco "discloses storing values from input fields in an association with an identifier for those fields . . . this is a distinct file independent of the screen layout and independent of any location where the fields may occur on the screen" (*id.*). More specifically, Appellants contend "it is seminal to the function and purpose of Di Franco that the data note [sic, not] be associated with a field having a known location because <u>Di Franco</u> seeks to fill the field of any form regardless of their position on the screen" (*id.*). Further, in Di Franco,

> if the data is tied to the particular location of the fields [it] would be unsuitable for population of either different versions of forms or forms where the locations of the fields has changed . . . [t]hus, the manner in which the Office attempts to combine the references is inappropriate and could only have been motivated by impermissible hindsight

(App. Br. 10–11).

Appellants further argue that "<u>Di Franco</u> saves the content of the form input fields and associates it with a tag so that when that input field is identified on a subsequent screen the data is copies [sic, copied] into the input field," and thus does not require, teach or suggest "a return of the internal state of the software function with which an input field is associated as a basis for repopulation of the field." Rather, Di Franco "merely teaches copying of data from one saved file to another" (App. Br. 11–12). Finally, Appellants argue that "Di Franco is expressly intended to address the GUI forms, which may be provided over the Internet from a third party, in other

words, the user or populator has no control," wherein, "to be useful, the data must be separate from the screen layout" (App. Br. 12).

We have considered all of Appellants' arguments and evidence presented. However, we disagree with Appellants' contentions regarding the Examiner's rejections of the claims. We agree with the Examiner's findings, and find no error with the Examiner's conclusion that the claims would have been obvious over the combined teachings.

We first note that Appellants do not specifically argue that cited references lack the claimed features. Instead, Appellants' arguments are directed to that the proposed combination of Di Franco with Dando and Powers would render Di Franco unsatisfactory for its intended purpose and/or would require impermissible hindsight (App. Br. 10–12).

Di Franco discloses that a user "can fill in a form with data the user selects and then store that selected data in a data storage file and then when the user requires, that selected data can be used to repopulate a form . . . at any time" (FF 1). No part of this disclosure requires that the form be different. Specifically, no portion of Di Franco states that including additional information, whether field location or any other data, in the data storage file, would prevent the repopulation of a form. Further, Appellants have not cited to any portion of Di Franco that suggests that including the location of a field in a saved file including input field values would render the Di Franco disclosure unsuitable for its intended purpose.

Mere attorney arguments that are unsupported by factual evidence are entitled to little probative value. *In re Geisler*, 116 F.3d 1465, 1470 (Fed. Cir. 1997); *see also In re De Blauwe*, 736 F.2d 699, 705 (Fed. Cir. 1984); *Ex parte Belinne*, No. 2009-004693, 2009 WL 2477843, slip op. at *7–8 (BPAI Aug. 10, 2009) (informative).

6

Although Appellants contend "it is seminal to the function and purpose of <u>Di Franco</u> that the data note [sic, not] be associated with a field having a known location because <u>Di Franco</u> seeks to fill the field of any form regardless of their position on the screen" (App. Br. 10), Appellants appear to view the combination in a different perspective than that of the Examiner. The issue here is not whether the skilled artisan would have bodily incorporated the entire system of Di Franco into the entire system of Dando, to provide *exclusively* user input field data tracking. Rather, the issue is whether the ordinarily skilled artisan, upon reading Dando and Di Franco, would have found it obvious to provide for the tracking of user input field data between times a user accesses the field, as shown in Di Franco, to the method and system providing for the storage of data related to GUI component location, as disclosed in Dando.

As the Examiner points out, the rejection "is based on a combination of Di Franco with Powers and Dando, both of which teach aspects of saving the screen layout of a user interface" (Ans. 2). Further, "both Di Franco and Dando teach that they record the identifiers of the user interface components" (Ans. 2). We, therefore, agree with the Examiner that "Di Franco does not require recording the locations of the components, nor does it preclude such" (Ans. 2–3), i.e., that Di Franco does not criticize, discredit, or otherwise discourage saving location data. *In re Fulton*, 391 F.3d 1195, 1201 (Fed. Cir. 2004).

The Supreme Court guides that the conclusion of obviousness can be based on the background knowledge possessed by a person having ordinary skill in the art. *KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398, 418 (2007). Here, Appellants have presented no evidence that providing a means of saving both location and user input field data related to each GUI element

7

would have been "uniquely challenging or difficult for one of ordinary skill in the art." *Leapfrog Enters., Inc. v. Fisher-Price, Inc.*, 485 F.3d 1157, 1162 (Fed. Cir. 2007) (*citing KSR,* 550 U.S. at 418).

Thus, we conclude that it would have been well within the skill of one skilled in the art to combine Di Franco's teachings with that of Dando in the same field of endeavor. We agree with the Examiner that

> [a]lthough the invention of Di Franco does not require recording the locations of the components, nor does it preclude such. The fact that the proposed combination would restore the location of the fields on the screen would not hinder the operation of matching the field identifiers with the saved input data.

(Ans. 2–3). Such a substitution would have been well within the skill of the art. *See KSR*, 550 U.S. at 417. That is, the skilled artisan is "a person of ordinary creativity, not an automaton." *KSR*, 550 U.S. at 421. We agree with the Examiner that Appellants' invention is simply a modification of familiar prior art teachings (as taught or suggested by the cited references) that would have realized a predictable result (*id.*). Minor differences between the prior art and a claimed device may be a matter of design choice absent evidence to the contrary (*see In re Rice*, 341 F.2d 309, 314 (CCPA 1965)).

We are also not persuaded that Di Franco's disclosed embodiment related to "GUI forms, which may be provided over the Internet from a third party" necessarily results in "the user or [the] populator [having] no control" (App. Br. 12). We agree with the Examiner, citing to paragraph 92, that Di Franco "teaches that the user is integrating the functionality into 'their' [the user's] GUI application window, which suggests a fair degree of control over the forms" (Ans. 3). We find this response persuasive, and further note

that as Di Franco states that the user "has the ability to save display values from *any* form at *any* point in time" (FF 1, emphasis added).

Finally, we must give the claim its broadest reasonable interpretation consistent with the Specification. *See In re Morris*, 127 F.3d 1048, 1054 (Fed. Cir. 1997). Although Appellants argue that Di Franco does not require, teach or suggest "a return of the internal state of the software function with which an input field is associated" (App. Br. 11–12), in claim 1, the term "internal state for each software function" is specifically defined by the claim language to be (1) the defined location and (2) the current content of the user input fields. A review of the Specification finds no definition of the "internal state for each software function" more specific than that defined in the claims. We do not find the Examiner's response that the broadest reasonable interpretation of a software function includes "the functions ('components') behind the various user interface[s], as taught by the cited references" is misplaced (Ans. 3). Support for this position is found in Dando, which states that a "view component displays the information of a particular *application* and provides an area where a user enters data and *executes action* on that data" (FF 2, emphasis added). Executing action on data entered into an area of an application suggests that a software function acts on that data (*id.*).

Based on the record before us, we find no error in the Examiner's rejection of independent claim 1 over the combination of Dando, Powers, and Di Franco.

As for claim 4, Appellants' argument that "[Di Franco] uses a discrete data file that is distinct from any screen layout or the form that is to be populated by the values saved therein" (App. Br. 12–13). Further, with respect to claim 14, Appellants argue that in Di Franco, "it is essential that

that data be separate and merely tagged with the appropriate fields, so that they can be appropriately populated regardless of their location . . . ." (App. Br. 13–14). These arguments largely reflect the arguments presented with respect to claim 1, and we incorporate our analysis herein, finding no error with the Examiner's rejection of claims 4 and 14 over the combination of Dando, Powers, and Di Franco.

As for claim 9, Appellants argue that Di Franco "neither teaches nor suggests the creation of the data structure containing these features nor does it teach or suggest the processing of a static data structure to recreate the display and the contents of user input fields . . . the Examiner has misinterpreted 'software object' as it being a static data structure" (App. Br. 14–15). The Examiner responds, and we agree, that the "plain and ordinary meaning of ['static data structure'] is a data structure of fixed size . . . in object oriented programming, data structures are implemented as objects" (Ans. 4).

A review of the Specification discloses no specific definition of a "static data structure" inconsistent with its plain meaning. More specifically, at paragraphs 13 and 28, XML strings are noted as exemplary static data structures, and that when "an embodiment of the invention collects information about software application and user interface state, it may place it in a 'persistable' or 'static' form. In other words, the information is formatted so that it may conveniently be stored . . ." (*id*. at 28). Accordingly, we find no error in the Examiner's broad but reasonable interpretation that a static data structure may encompass a "container state object" as disclosed by Dando (FF 2).

As for claim 12, Appellants argue that

10

> Tafoya relates to completion of partial information as a user begins to type . . . predicts what the remainder will be from data captured from a plurality of sources, *not limited to the historical entry in the particular field*. Accordingly, a suggestion by the Examiner that Tafoya in any way teaches or suggests an explicit history of a particular field is factually incorrect.

App. Br. 15, emphasis added. The Examiner responds by noting that "Tafoya teaches suggesting previous form entries . . . further suggestions may come from other sources . . ." (Ans. 4–5).

Tafoya discloses a user input field for entering e-mail addresses used to send e-mails, and includes a drop-down menu including previous entries (Fig. 6; col. 2, ll. 17–25). The previous entries may be tracked from a user's previous entries when sending or receiving e-mail (*id.*). We find no error with the Examiner's reliance on Tafoya for disclosing and suggesting "states of the software components *include* a history of entries of an input field of the display screen" (claim 12, emphasis added). Accordingly, we find no error in the Examiner's rejection of claim 12 over the combination of Dando, Di Franco, and Tafoya.

Appellants do not provide arguments for claims 3, 5–8, and 13–19, and thus, we also affirm the rejection of claims 3, 5–8, 10, and 14–18 over Dando, Powers. and Di Franco; of claim 5 over Dando, Powers, Di Franco, and Le Hors; of claim 13 over Dando, Di Franco, and Cutler; and of claim 19 over Dando, Di Franco, Powers, and Official Notice.

V. CONCLUSION AND DECISION

We affirm the Examiner's rejections of claims 1, 3–10, and 12–19 under 35 U.S.C. § 103(a).[1]

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

<u>AFFIRMED</u>

---

[1] In the event of further prosecution of this application, we leave it to the Examiner to evaluate the claims for compliance with 35 U.S.C. § 101 in view of the Supreme Court decision in *Bilski v. Kappos*, 130 S. Ct. 3218, 3221 (2010), *Manual of Patent Examining Procedure* (MPEP) revised § 2106.01 (9th Ed., Rev. 9, Nov. 2015), and post-*Bilski* application under § 101, including *CyberSource Corp. v. Retail Decisions, Inc.*, 654 F.3d 1366, 1371 (Fed. Cir. 2011) (holding that a method for verifying the validity of a credit card transaction over the Internet to be nonstatutory as an abstract idea capable of being performed in the human mind or by a human using a pen and paper). In particular, we leave it to the Examiner to evaluate whether claim 1's "converting" and "recreating" of a screen layout constitutes an abstract idea. We further refer to *Content Extraction*, where the Federal Circuit has provided additional guidance on the issue of statutory subject matter by holding claims to collecting data, recognizing certain data within the collected data set, and storing that recognized data in memory, was not tied to a specific structure or machine, and was thus an abstract idea and ineligible under § 101. *Content Extraction & Transmission LLC v. Wells Fargo Bank, N.A.*, 776 F.3d 1343 (Fed. Cir. 2014). Abstract ideas have been identified by the courts by way of example, including fundamental economic practices, certain methods of organizing human activities, an idea "of itself," and mathematical relationships/ formulas. *Alice Corp. Pty. Ltd. v. CLS Bank Int'l*, 134 S. Ct. 2347, 2355–56 (2014). Although the Board is authorized to reject claims under 37 C.F.R. § 41.50(b), no inference should be drawn when the Board elects not to do so. *See* MPEP § 1213.02.