Trials@uspto.gov
Tel: 571-272-7822

Paper 9
Entered: August 12, 2013

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

SAS INSTITUTE, INC.
Petitioner

v.

COMPLEMENTSOFT, LLC
Patent Owner

Case IPR2013-00226
Patent 7,110,936 B2

Before KEVIN F. TURNER, JUSTIN T. ARBES, and JENNIFER S. BISK, *Administrative Patent Judges.*

BISK, *Administrative Patent Judge*.

DECISION
Institution of *Inter Partes* Review
37 C.F.R. § 42.108

I. INTRODUCTION

A. Background

SAS Institute, Inc. ("SAS") filed a petition ("Pet.") (Paper 1) to institute an *inter partes* review of claims 1-16 of Patent 7,110,936 B2 (the "'936 patent") pursuant to 35 U.S.C. § 311 *et seq*. ComplementSoft, LLC ("ComplementSoft") filed a preliminary response ("Prelim. Resp.") (Paper 8). We have jurisdiction under 35 U.S.C. § 314. We conclude that SAS has satisfied the burden to show, under 35 U.S.C. § 314(a), that there is a reasonable likelihood that it would prevail with respect to at least one of the challenged claims.

SAS contends that the challenged claims are unpatentable under 35 U.S.C. §§ 102 and/or 103 based on the following specific grounds (Pet. 11-12):[1]

[1] SAS also asserts that "[t]o the extent not explicitly enumerated above, claims 2-16 are unpatentable over each reference and combination of references asserted for claim 1 in view of the prior art." Pet. 12. This assertion fails to satisfy the requirement that a petition must identify with particularity each claim challenged, the grounds on which the challenge is based, and the evidence that supports the grounds for the challenge to each claim. *See* 35 U.S.C. § 312(a)(3); 37 C.F.R. §§ 42.22(a), 42.104(b)(4)-(5). We, therefore, do not further address these unsupported challenges.

Reference[s] [2]	Basis	Claims challenged
Coad	§ 102	1
Coad, Oracle Primer, and Oracle8 Primer	§ 103	1
Antis	§ 102	1-3 and 5
Antis and Coad	§ 103	1-3, 5, 6, 8, 10-12, 15, and 16
Antis, Coad, and Burkwald	§ 103	4
Antis, Coad, and Eick	§ 103	7
Antis, Coad, and Building Applications	§ 103	9
Antis, Coad, and Corda	§ 103	13
Antis, Coad, and Access 97 Visual Basic	§ 103	14

For the reasons described below, we institute an *inter partes* review of claims 1 and 3-10 based on the following grounds: (1) claim 1 is obvious over Coad combined with Oracle Primer and Oracle8 Primer; (2) claims 1, 3, 5, 6, 8, and 10 are obvious over Antis combined with Coad; (3) claim 4 is obvious over Antis combined with Coad and Burkwald; (4) claim 7 is obvious over Antis combined with Coad and Eick; and (5) claim 9 is obvious over Antis combined with Coad and Building Applications.

We decline to institute *inter partes* review of (1) claims 2 or claims 11-16;

[2] U.S. Patent 5,572,650 (Ex. 1005) ("Antis"); U.S. Patent 6,851,107 (Ex. 1006) ("Coad"); U.S. Patent 6,356,285 (Ex. 1007) ("Burkwald"); U.S. Patent 5,937,064 (Ex. 1008) ("Eick"); Evan Callahan, MICROSOFT ACCESS 97 VISUAL BASIC STEP BY STEP (1997) (Ex. 1009) ("Access 97 Visual Basic"); U.S. Patent 5,782,122 (Ex. 1010) ("Corda"); Microsoft Corporation, BUILDING APPLICATIONS WITH MICROSOFT ACCESS 97 (1996) (Ex. 1011) ("Building Applications"); Rajshekhar Sunderraman, ORACLE PROGRAMMING: A PRIMER (1999) (Ex. 1012) ("Oracle Primer"); and Rajshekhar Sunderraman, ORACLE8 PROGRAMMING: A PRIMER (2000) (Ex. 1013) ("Oracle8 Primer").

(2) claim 1 based on anticipation by Coad; or (3) claims 1, 3, or 5 based on anticipation by Antis.

B. The Invention

The '936 patent describes a language independent software development tool having a graphical user interface, also referred to as an Integrated Development Environment or IDE. Ex. 1001, col. 1, ll. 15-19. In particular, the patent describes an IDE for exchanging, editing, debugging, visualizing, and developing software code for "data manipulation centric languages." *Id.* at col. 1, l. 64 – col. 2, l. 3.

The Summary of the Invention describes the IDE as including each of the following: (1) a document manager that manages connections between computers and transfers files (col. 2, ll. 20-26); (2) an editor that can modify code within an existing file using advanced editing features or create a new file (col. 2, ll. 27-33); (3) a visualizer that generates a graphical representation of the program flow, data flow, or logic of the code (col. 2, ll. 34-49); (4) a template manager that allows the user to browse through a repository of existing code or templates and copy a selected template into a file for editing (col. 2, ll. 50-54); and (5) a parser layer that detects the type of code in the selected file and activates the corresponding rules and logic (col. 2, ll. 55-62).

Claim 1, reproduced below, is the '936 patent's only independent claim:

1. An integrated development environment, comprising:

a document manager for retrieving source code programmed using one of a plurality of types of data manipulation languages;

an editor for displaying the retrieved source code and providing a means for a user to edit the retrieved source code;

a parser layer which detects the one of the plurality of types of data manipulation languages in which the retrieved source code is

> programmed and which activates rules and logic applicable to the detected one of the plurality of types of data manipulation languages; and
>
> a visualizer dynamically linked to the editor for displaying graphical representations of flows within the retrieved source code using the rules and logic applicable to the detected one of the plurality of types of data manipulation languages and activated by the parser,
>
> wherein the editor, parser layer and visualizer cooperate such that edits made to the source code using the editor are automatically reflected in the graphical representations of flows displayed by the visualizer and edits made to the graphical representations of flows in the visualizer are automatically reflected in the source code displayed by the editor.

We note that the '936 patent is asserted currently in *ComplementSoft, LLC v. SAS Institute, Inc.*, Docket No. 1:12-cv-07372 (N.D. Ill. Sept. 14, 2012) ("the related litigation"). *See* Pet. 58; Paper 6 at 2.

C. Claim Construction

As a step in our analysis for determining whether to institute a trial, we determine the meaning of the claims. Consistent with the statute and the legislative history of the America Invents Act (AIA), the Board will interpret claims using the broadest reasonable construction in light of the specification. *See* Office Patent Trial Practice Guide, 77 Fed. Reg. 48756, 48766 (Aug. 14, 2012); 37 CFR § 42.100(b). Both parties submit proposed constructions for several claim terms. Pet. 12-14; Prelim. Resp. 10-15. We summarize each of the proposed interpretations below:

Claim Term	**SAS Proposal**	**ComplementSoft Proposal**
automatically [reflected]	without user intervention	generated by the IDE, not by the user
data	a programming language	a computer programming language

manipulation language	used to access data in a database, such as to retrieve, insert, delete, or modify data in a database	that enables a programmer to create a datacentric program
integrated development environment		a single comprehensive software development tool capable of assisting users in the editing, visualizing, debugging, and development of software
editor		a component of the IDE that can create new source code files, and also display and modify source code within existing source code files
graphical representation of flows		a diagram using icons and arrows to depict procedures in the order they occur in a data manipulation language and/or the movement of data through the processes performed by the procedures in the order they occur

We have considered the parties' proposals, but conclude that only the terms "data manipulation language" and "graphical representation of flows" require an explicit construction for purposes of this decision.

1. Data Manipulation Language

Both parties offer a proposed interpretation of the claim term "data manipulation language." SAS asserts that the term means "a programming language used to access data in a database, such as to retrieve, insert, delete, or modify data in the database." Pet. 14. SAS bases this proposed interpretation on the '936 patent's disclosure of SQL® and Oracle® RDBMS as data manipulation languages for accessing data in a database. *Id.* (citing Ex. 1001, col. 1, ll. 20-25).

ComplementSoft argues that SAS's proposed definition is "inadequate" in

that it "only provides examples of what a data manipulation language can do, not what a data manipulation language is." Prelim. Resp. 10-11. ComplementSoft proposes instead that "data manipulation language" means "a computer programming language that enables a programmer to create a datacentric program," i.e., a program in which "the data drives the objectives." *Id.* at 10. As support for this definition, ComplementSoft cites to the same language in the specification as does SAS. *Id.* (citing Ex. 1001, col. 1, ll. 20-30). In addition, ComplementSoft points to several other sections of the Specification, including language stating that the preferred embodiment is designed to work with specific types of code (col. 9, ll. 47-53; col. 17, ll. 40-45), language referring to "data processing steps" (col. 2, ll. 40-42), and figures illustrating data accessing and processing (Figs. 9 and 17).

We are persuaded that the definition proposed by SAS is the broadest reasonable construction of the term. The phrase "data manipulation language" is not defined explicitly in the written description of the '936 patent. Thus, there is a "heavy presumption" that the term carries its ordinary and customary meaning. *CCS Fitness, Inc. v. Brunswick Corp.*, 288 F.3d 1359, 1366 (Fed. Cir. 2002). We are persuaded that SAS's proposed definition is consistent with the ordinary and customary meaning of the term—"a language that is used to insert data in, update, and query a database." MICROSOFT COMPUTER DICTIONARY at 125 (4th ed. 1999).

We are not persuaded by ComplementSoft's arguments for its proposed interpretation. First, *both* proposed definitions define the term by describing what a "data manipulation language can do": SAS's proposal is that the language can be "used to access data in a database, such as to retrieve, insert, delete, or modify data in the database," while ComplementSoft proposes that the language "enables a programmer to create a datacentric program." Second, none of the citations to the

'936 patent relied upon by ComplementSoft mandate its more narrow interpretation. The citations pointed to by ComplementSoft also support SAS's broader proposed definition.

Thus, for purposes of this decision, we construe the claim term "data manipulation language" to be a programming language used to access data in a database, such as to retrieve, insert, delete, or modify data in the database.

2. Graphical Representation of Flows

SAS does not address explicitly the construction of the claim term "graphical representation of flows." ComplementSoft asserts that the term should be defined as "a diagram using icons and arrows to depict procedures in the order they occur in a data manipulation language and/or the movement of data through the processes performed by the procedures in the order they occur." Prelim. Resp. 13-14. We are not persuaded that this is the broadest reasonable interpretation of the term.

First, we are not persuaded that the term "graphical representation" is limited to "icons and arrows" as proposed by ComplementSoft. *See id.* The plain and ordinary meaning of the word "graphical representation" is using a picture or graph to depict something else. *See, e.g.,* AMERICAN HERITAGE DICTIONARY at 573, 1049 (2nd College Ed. 1982) (defining "graphical" as "of or pertaining to pictorial representation"). Thus, the plain and ordinary meaning of "graphical representation of flows" is a picture or graph that depicts flows. Although the '936 patent describes the use of icons and arrows in diagrams (*see, e.g.,* col. 2, ll. 38-42), ComplementSoft does not point to language in the patent that limits the diagrams to those particular symbols.

Second, we are not persuaded that the "flows" are limited to "procedures in the order they occur in a data manipulation language and/or the movement of data

through the processes performed by the procedures in the order they occur" as asserted by ComplementSoft. *See* Prelim. Resp. 13-14. The plain and ordinary meaning of the term "flow" in the context of computer software is a map of the progression (or path) through the executing source code.[3] The '936 patent describes two kinds of flows, "program flows" and "data flows." Ex. 1001, col. 2, ll. 38-42. For example, a "program flow diagram" depicts a map of the progression of control through the executing source code and a "data flow diagram" depicts a map of the path of data through the executing source code. Ex. 1001, col. 2, ll. 33-43. ComplementSoft, however, does not point to language in the patent that limits the term "flows" to only those examples.

For purposes of this decision, therefore, we are persuaded that the broadest reasonable interpretation of the term "graphical representation of flows" is a diagram that depicts a map of the progression (or path) through the source code.

3. Means-Plus-Function Limitations

Several of the challenged claims include the language "means" or "means for" and therefore are presumed to invoke 35 U.S.C. § 112 ¶ 6.[4] *Personalized Media Commc'ns LLC v. Int'l Trade Comm'n*, 161 F.3d 696, 703-04 (Fed. Cir. 1998). This presumption is not conclusive. *Sage Prods., Inc. v. Devon Indus., Inc.*, 126 F.3d 1420, 1427-28 (Fed. Cir. 1997). For example, section 112 is not implicated where a claim uses the word "means" but does not specify a corresponding function. *Id.* at 1427-28. Section 112 also is not implicated where a

[3] See, for example, flowchart: "A graphic map of the path of control or data through the operations in a program or an information-handling system." MICROSOFT COMPUTER DICTIONARY at 190 (4th ed. 1999).
[4] Section 4(c) of the AIA re-designated 35 U.S.C. § 112 ¶ 6, as 35 U.S.C. § 112(f). Because the '936 patent has a filing date before September 16, 2012 (effective date), we will refer to the pre-AIA version of 35 U.S.C. § 112.

claim recites a corresponding function, but the claim also recites sufficient structure, material, or acts to perform entirely the recited function. *Id.*

Claim 1 recites "an editor for displaying the retrieved source code and providing a *means for a user to edit* the retrieved source code" (emphasis added). SAS does not propose, specifically, a construction for this limitation. ComplementSoft asserts that this language is not a statutory means-plus-function clause because it "provides further definition for the functionality of the editor." Prelim. Resp. 11. ComplementSoft's argument appears to be that the recited editor provides sufficient structure to perform entirely the recited function – editing the retrieved source code – and, therefore, this limitation does not implicate section 112. This is consistent with ComplementSoft's proposed interpretation of the claim term "editor" as summarized above. On this record, we conclude that ComplementSoft's argument is reasonable, and do not interpret the phrase "means for a user to edit" to be a means-plus-function limitation.

Claim 11 depends from claim 1 and further recites "a *means for allowing* the source code to be executed both locally and remotely" (emphasis added). Neither party proposes an interpretation for this limitation. Because claim 11 uses the words "means for" modified by functional language and the limitation is not modified by any structure recited in the claim to perform the claimed function – allowing source code to be executed both locally and remotely – we interpret this limitation to be a means-plus-function limitation. As discussed in more detail below, SAS did not meet its burden to identify how claim 11 is to be construed. *See* 37 C.F.R. § 42.104(b)(3). Specifically, SAS did not address the corresponding structure in the Specification for the means-plus-function limitation.

II. ANALYSIS

A. Claims 11-16

SAS has the burden to establish a reasonable likelihood of prevailing on its assertion that claim 11, and those claims that depend from claim 11—claims 12-16—are unpatentable over the asserted prior art. An essential part of that showing is identifying how each challenged claim is to be construed. 37 C.F.R. § 42.104(b)(3). Specifically, the rules require that "[w]here the claim to be construed contains a means-plus-function or step-plus-function limitation[,] . . . the construction of the claim must identify the specific portions of the specification that describe the structure, material, or acts corresponding to each function." *Id.* As discussed above, SAS does not identify what structure in the Specification it believes corresponds to the means-plus-function limitation of claim 11. This failure is fatal to SAS's challenge of claims that include that limitation. Indeed, SAS's discussion of the asserted prior art (Coad and Antis) in relation to claim 11, as well as the analysis of Dr. Roussopoulos, does not address any corresponding structure in the Specification of the '936 patent. *See* Pet. 47-48 (citing Ex. 1015, ¶¶ 191-95). SAS's analysis, therefore, is insufficient to show that the prior art teaches the means-plus-function limitation of claim 11. Thus, we decline to institute *inter partes* review on any proposed ground for claims 11-16.

B. Coad

1. Overview of Coad

Coad discloses a software development tool that allows a developer to view and modify simultaneously textual and graphical displays of source code regardless of the programming language in which the code is written. Ex. 1006, Abstract, col. 4, ll. 38-41. In the Background of the Invention, Coad describes conventional software development tools that allow the user to view Unified

Modeling Language (UML)—a graphical representation or model using object oriented design—and source code at the same time. *Id.* at col. 1, l. 47 – col. 2, l. 22. Coad lists several disadvantages of these prior art systems, including that the files containing the source code and UML are not synchronized and that the tools work with only a single programming language. *Id.* at col. 2, ll. 22-36. "Thus, a tool **100** that is designed for Java™ programs cannot be utilized to develop a program in C++." *Id.* at col. 2, ll. 36-40.

Coad describes using the software development tool to: (1) open a file that contains existing source code or create a file in which source code will be developed (col. 15, ll. 60-64); (2) modify existing source code using an incremental code editor (ICE) (col. 4, ll. 54-58); (3) view several models of the source code using static, dynamic, and functional diagrams (col. 16, l. 58 – col. 17 l. 32); (4) obtain templates for the current programming language (col. 16, ll. 4-9); and (5) convert source code into the language-neutral representation for viewing and vice versa using a parser layer (col. 5, ll. 50-55; col. 16, ll. 4-16).

2. Anticipation of Claim 1

SAS asserts that Coad anticipates claim 1. Pet. 25-31. In particular, SAS asserts that Coad discloses the limitation of "source code programmed using one of a plurality of types of data manipulation languages" because it discusses the use of "a plurality of types of programming languages," including C++ and Java. Ex. 1006, Abstract; col. 16, ll. 1-4. SAS also provides the testimony of Dr. Nick Roussopoulos stating that Java and C++ "had functions and structures through the use of/embedding of SQL statements that allowed the programming language to access data, such as to retrieve, insert, delete, or modify data in a database." Ex. 1015, ¶ 49. Dr. Roussopoulos testified that "a well-known product (Oracle) allowed a Java program to use SQL to retrieve data query results." *Id.* at ¶ 50. Dr.

Roussopoulos, however, did not testify that all versions of C++ or Java included data manipulation functionality or that Coad's disclosure necessarily included data manipulation languages.

ComplementSoft responds that Coad does not meet its burden of showing anticipation of the required data manipulation language limitation by showing that Coad either explicitly or inherently discloses the use of data manipulation languages. Prelim. Resp. 27. We agree with ComplementSoft. Coad does not explicitly disclose that any of the programming languages referred to in the specification include data manipulation capabilities. Further, SAS does not meet its burden to prove inherency by showing that such capability is inherent in Coad's disclosure. Therefore, we are not persuaded that SAS has shown a reasonable likelihood of prevailing in its assertion that claim 1 is anticipated by Coad.

3. Obviousness of Claim 1 over Coad, Oracle Primer, and Oracle8 Primer

SAS asserts that claim 1 is obvious over Coad combined with Oracle Primer and Oracle8 Primer. SAS relies on the Oracle documents for describing the use of SQL within Java and C++ and thus disclosing the data manipulation language limitation. Pet. 31-32 (citing Ex. 1015, ¶¶ 111-115). SAS points to Figures 11-17 of Coad as depicting aspects of the view for displaying graphical representations of flows in source code. Pet. 29-30. In addition, SAS asserts, and ComplementSoft does not dispute, that in the related district court litigation, ComplementSoft conceded that Coad discloses the "editor" limitation. Pet. 27 (citing Ex. 1016 (ComplementSoft's Response to SAS's Invalidity Contentions) at 18).

ComplementSoft argues that the asserted combination of references does not

meet the required "graphical displays of flows"[5] limitation. Prelim. Resp. 28. According to ComplementSoft, the focus in Coad on object-oriented languages, and an incompatibility of the treatment of data between object-oriented languages and relational databases, means that the graphical representations of flows in Coad are incompatible with those claimed in the '936 patent. Prelim. Resp. 28-32. We are not persuaded by this argument. Figure 14 of Coad, pointed to by SAS (Pet. 28-30), "displays a sequence diagram of source code." Ex. 1006, col. 4, ll. 16-18. As described in Coad, in a sequence diagram, "the vertical dimension represents time" and the diagram depicts "the time ordering of messages along the vertical axis" representing "an interaction . . . to effect a desired operation or result." *Id.* at col. 17, ll. 1-15. Figure 14, therefore, depicts a step-by-step progression through the source code. *See id.* at Fig. 14. More specifically, Figure 14 depicts the source code program's path of control from one step to another through the program—a program flow diagram. Thus, Coad discloses "graphical representations of flows" as we interpret the term.

ComplementSoft also argues that the Oracle Primers are non-analogous art to Coad. Prelim. Resp. 33. According to ComplementSoft, Coad's field of endeavor is IDEs and that of the Oracle Primers is introductory texts for SQL programming of Oracle. *Id.* ComplementSoft asserts that these fields are not analogous. *Id.* ComplementSoft adds that the two references also do not relate to the same problem of improving IDEs. *Id.* SAS, however, asserts that all three references are directed to computer programming, generally, and to the Java and C++ programming languages, specifically. Pet. 25. We agree with SAS that the

[5] Because none of the claims include this particular language, we assume that ComplementSoft is referring to the limitation "graphical representations of flows."

references have similar purposes and overlapping teachings and all relate to software development using Java and C++. We also find reasonable SAS's rationale that a person of ordinary skill would have combined the teachings of these references in order to enhance the utility of the programming environment to include data manipulation. *See* Pet. 25.

We are persuaded that SAS has shown a reasonable likelihood of prevailing in its assertion that claim 1 is obvious over Coad combined with Oracle Primer and Oracle8 Primer.

4. Previous Office Consideration of Coad

Finally, we note that Coad was applied as a prior art reference during prosecution of the '936 patent. *See, e.g.*, Ex. 1002.[6] The Oracle Primers—and the specific combination of Coad and the Oracle Primers asserted by SAS—however, were not considered. While we are mindful of the burden on ComplementSoft and the Office in analyzing previously considered prior art, substantially the same prior art and arguments were not before the Office previously. *See* 35 U.S.C. § 325(d). Moreover, for the reasons explained above, we conclude that SAS's arguments based on the combination of Coad and the Oracle Primers have merit.

C. Antis

1. Overview of Antis

Antis relates to visually displaying structural characteristics of a large database for development purposes. Ex. 1005, Abstract. Antis describes a long felt need for a tool to display the characteristics of a database without semantic information such that explicit and implicit data structures can readily be observed

[6] This exhibit is not marked with individual page numbers, which is a violation of 37 C.F.R. § 42.63(d)(2)(i).

to facilitate use, development, and maintenance of large databases. *Id.* at col. 2, ll. 25-29. Antis solves this problem by displaying statistics and characteristics of an entire relational database in one overall view with semantic information separated out and shown in additional views that interactively are linked to the overall view and to each other. *Id.* at ll. 31-39.

Antis describes several specific views, including: (1) an "over view," the highest view level of the large relational database (col. 4, ll. 1-3); (2) a "specification view," a view of the actual specification(s) of the database in the database description language or languages (col. 5, ll. 4-8); (3) an "associations view" showing associations between a selected relation and other relations of the database through queries and other supported relational database management system (RDBMS) mechanisms (col. 5, ll. 32-36); (4) a "path view" presenting all of the shortest paths connecting any two selected relations (col. 6, ll. 43-47); (5) a "code view" that displays the application source code that uses the currently selected relation (col. 7, ll. 31-35); (6) a "layout view" showing the physical layout in memory of a tuple of a relation as well as the relative sizes of attributes of the relation (col. 8, ll. 5-9); and (7) a "domain view" that shows the domains used by a given relation and that is useful to the user for exploring how domains are used and shared among relations (col. 8, ll. 22-25).

2. Anticipation of Claims 1-3 and 5

SAS asserts that Antis anticipates claims 1-3 and 5. Pet. 32-41. Specifically, SAS asserts that the "code view" of Antis is used to edit retrieved source code. Pet. 34-35 (citing Ex. 1005, Fig. 12; col. 8, 63-66). According to SAS, "Antis discloses that changes made in any of the disclosed views (*e.g.*, edits to the source code made in the expanded code view of a definition of a new object) cause corresponding changes in the other views." Pet. 35 (citing Ex. 1005, col. 9,

ll. 17-28).

ComplementSoft argues that Antis fails to disclose the claimed "editor" because Antis does not disclose the capability of modifying code within files. Instead, the language relied upon by SAS discloses only that the expanded code view is used to view code, not actually change that code. ComplementSoft points out that although Antis discloses that a "change" in any one view will cause corresponding "changes" in other views, this does not mean necessarily that the underlying source code is changed, but instead could simply mean that what is shown in the view could change. Prelim. Resp. 34.

We agree with ComplementSoft. SAS does not point to any language in Antis stating explicitly that any underlying files are ever changed or that the computer system described is used for anything other than viewing the source code of the application. Consistent with this interpretation of Antis as a system for visually displaying a static view, the specification describes user inputs as "cursor touches" or "mouse button clicks." Ex. 1005, col. 9, ll. 30-35. For example, Antis states that the user can use the keyboard and mouse to "examine the results in more detail, or call up other linked displays to obtain more information" (col. 3, ll. 58-60), and "touching any code box with a cursor causes that box to be highlighted and its designation to be displayed" and "[c]licking the mouse button on a code box highlights in the over view all relations that use the corresponding unit of code" (col. 7, ll. 46-54). Antis does not, however, describe editing or modifying underlying source code files.

Thus, we agree with ComplementSoft that SAS has not shown a reasonable likelihood of prevailing in its assertion that claims 1-3 and 5, all of which require an editor, are anticipated by Antis.

3. Obviousness of Claims 1-3, 5, 6, 8, and 10 over Antis and Coad

SAS asserts that claims 1-3, 5, 6, 8, and 10 are obvious over Antis combined with Coad. Pet. 41-52. SAS points to Coad as disclosing an incremental code editor for displaying and editing retrieved source code. Pet. 41 (citing Ex. 1006, col. 4, ll. 54-60). ComplementSoft argues that incorporating the editor of Coad into Antis is not possible because Antis does not actually manipulate data. Prelim. Resp. 37. This argument is not persuasive. "It is well-established that a determination of obviousness based on teachings from multiple references does not require an actual, physical substitution of elements." *In re Mouttet*, 686 F.3d 1322, 1332 (Fed. Cir. 2012) (citing *In re Etter*, 756 F.2d 852, 859 (Fed. Cir. 1985) (en banc) (noting that the criterion for obviousness is not whether the references can be physically combined, but whether the claimed invention is rendered obvious by the teachings of the prior art as a whole)).

ComplementSoft also argues that neither reference on its own, nor the combination of the two references, teaches "graphical representations of flows" as required by the challenged claims. Prelim. Resp. 36-38. As discussed above, we are persuaded that Coad discloses this limitation. SAS explains that a person of ordinary skill in the art would have had a reason to combine Antis and Coad because they are both directed to software development tools that provide visual representations of source code. Pet. 18. The combination of the Coad with Antis would have allowed for easier source code debugging and a more accurate code view display according to SAS. *Id.* (citing Ex. 1015, ¶¶ 164-67). We are persuaded that this rationale is reasonable.

In summary, we have reviewed SAS's arguments in relation to each of the claims 1, 3, 5, 6, 8, and 10 and find that there is a reasonable likelihood that SAS will prevail in its challenge that these claims are obvious over a combination of

Antis and Coad.

Claim 2 depends from claim 1 and adds the additional limitation that "the graphical representations of flows depict data flows." As discussed above, we are persuaded that Figure 14 of Coad depicts the source code program's path of control from one step to another through the program—a program flow diagram. ComplementSoft argues that neither Antis nor Coad nor the combination of the two references discloses a "graphical representation" of a "data flow." Prelim. Resp. 38-40.

We agree with ComplementSoft. SAS points to the code view of Antis as disclosing this limitation because "the code view provides a visualization of data flows in the retrieved source by providing a visual representation of which pieces of source code access which data in the relational database." Pet. 39 (citing Ex. 1005, col. 7, ll. 31-45). We are not persuaded that this is equivalent to a depiction of a map of the path of data through the executing source code. SAS also points to Coad as disclosing this limitation, generally pointing to a description of all views of Coad. Pet. 43-44 (citing Ex. 1006, col. 16, l. 57 – col. 17, l. 47). It is unclear exactly which view SAS equates to the claimed "graphical representation" of a "data flow." Moreover, it is not clear on its face that Coad discloses this limitation as claimed. SAS has failed to demonstrate a reasonable likelihood that it would prevail on a challenge of claim 2 based on obviousness over Antis and Coad.

4. Obviousness of Claim 4 over Antis, Coad, and Burkwald

Claim 4 depends from claim 1 and adds the limitation that "the graphical representations of data flows are expandable and collapsible." SAS relies on Burkwald—a patent directed to a "system for visually representing modification information about a[] characteristic-dependent information processing system"—as disclosing this limitation. *See* Pet. 52 (citing Ex. 1007, col. 14, l. 43 – col. 15, l. 4);

Ex. 1007, Title. SAS explains that a person of ordinary skill in the art would have had a reason to combine the three references because they are all related to software development tools that provide visual representations of source code. Pet. 19. Burkwald's teaching of expanding and collapsing graphical representations of flows would have provided a developer with flexibility in the amount of detail shown in the view according to SAS. *Id.* at 19-20. ComplementSoft does not address this proposed ground of unpatentability.

We are persuaded that there is a reasonable likelihood that SAS will prevail in its challenge that claim 4 is obvious over a combination of Antis, Coad, and Burkwald.

5. Obviousness of Claim 7 over Antis, Coad, and Eick

Claim 7 depends from claim 6 and adds the limitation that "the document manager comprises a security layer for managing secure connections with the one or more remote computers." SAS relies on Eick—a patent directed to a "system and method for interactive visualization, analysis and control of a dynamic database"—as disclosing this limitation. *See* Pet. 53 (citing Ex. 1008, col. 4, ll. 19-27); Ex. 1008, Title. SAS explains that a person of ordinary skill in the art would have had a reason to combine the three references because they are all related to visual representations of source code and data structures. Pet. 21. Moreover, Eick's teaching of a security layer for managing secure connections with remote computers would allow the systems of Antis and Coad to be distributed to one or more locations without a substantial security risk according to SAS. *Id.* at 21-22. ComplementSoft does not address this proposed ground of unpatentability.

We are persuaded that there is a reasonable likelihood that SAS will prevail in its challenge that claim 7 is obvious over a combination of Antis, Coad, and Eick.

6. Obviousness of Claim 9 over Antis, Coad, and Building Applications

Claim 9 depends from claim 8 and adds the limitation that "the template manager is adapted to automatically correct segments of the source code." SAS relies on Building Applications—a book including information about Microsoft Access 97 software—as disclosing this limitation. Pet. 54-55 (citing Ex. 1011, pp. 52-54). SAS explains that a person of ordinary skill in the art would have had a reason to combine the three references because they are all related to software development tools that provide visual representations of source code. Pet. 23. Building Applications's teaching of automatically correcting segments of source code determined to have errors would simplify the debugging of source code in Antis and Coad according to SAS. *Id.* at 23-24. ComplementSoft does not address this proposed ground of unpatentability.

We are persuaded that there is a reasonable likelihood that SAS will prevail in its challenge that claim 9 is obvious over a combination of Antis, Coad, and Building Applications.

III. CONCLUSION

We institute an *inter partes* review of claims 1 and 3-10 based on the following grounds under 35 U.S.C. § 103(a): (1) claim 1 is obvious over Coad combined with Oracle Primer and Oracle8 Primer; (2) claims 1, 3, 5, 6, 8, and 10 are obvious over Antis combined with Coad; (3) claim 4 is obvious over Antis combined with Coad and Burkwald; (4) claim 7 is obvious over Antis combined with Coad and Eick; and (5) claim 9 is obvious over Antis combined with Coad and Building Applications.

We decline to institute *inter partes* review of (1) claims 2 or claims 11-16; (2) claim 1 based on anticipation by Coad; or (3) claims 1, 3, or 5 based on

anticipation by Antis.

IV. ORDER

For the reasons given, it is

ORDERED that the Petition is granted as to claims 1 and 3-10.

FURTHER ORDERED that pursuant to 35 U.S.C. § 314(a), *inter partes* review of the '936 patent is hereby instituted commencing on the entry date of this Order, and pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is hereby given of the institution of a trial.

FURTHER ORDERED that the trial is limited to the grounds and claims listed in the Conclusion. No other grounds are authorized as to these claims.

FURTHER ORDERED that an initial conference call with the Board is scheduled for 2 PM Eastern Time on September 12, 2013. The parties are directed to the Office Trial Practice Guide, 77 Fed. Reg. 48756, 48765-66 (Aug. 14, 2012) for guidance in preparing for the initial conference call, and should come prepared to discuss any proposed changes to the Scheduling Order entered herewith and any motions the parties anticipate filing during the trial.

PETITIONER:

David B. Cochran
John V. Biernacki
John A. Marlott
JONES DAY
dcochran@jonesday.com
jvbiernacki@jonesday.com
jamarlott@jonesday.com

PATENT OWNER:

James Hanft
George Yu
SCHIFF HARDIN LLP
jhanft@schiffhardin.com
gyu@schiffhardin.com