

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TABLEAU SOFTWARE, LLC
Petitioner

v.

ICHARTS LLC
Patent Owner

Case IPR2024-01389
Patent No. 8,520,000

**DECLARATION OF CHRISTOPHER SCHMANDT, IN SUPPORT OF
PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 8,520,000**

Mail Stop “PATENT BOARD”

Patent Trial and Appeal Board
U.S. Patent & Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

Tableau EX1003
IPR2024-01389
U.S. Patent No. 8,520,000

TABLE OF CONTENTS

I.	Introduction	1
II.	List of Documents Considered	2
III.	Qualifications.....	8
IV.	Legal principles	11
A.	Claim Construction	12
B.	Obviousness.....	12
C.	Written Description.....	14
D.	A Person of Ordinary Skill in the Art.....	15
V.	U.S. Patent No. 8,520,000.....	16
A.	Claims.....	26
B.	Prosecution History of the '000 Patent.....	28
C.	Provisional Application No. 61/077,728	40
D.	New Matter in U.S. Application No. 12/372,696	45
E.	Level of Ordinary Skill in the Art	47
VI.	Technology Overview.....	47
A.	Data Visualization Tools.....	47
B.	Flash	64
C.	JavaScript	66
D.	Drag and Drop	68
E.	Panes	71
VII.	Overview of the Prior Art	75

A.	Rostoker discloses a platform for generating, modifying, configuring or adapting interactive charts.	75
B.	Strong discloses distributing a data object upon modification to the data object.....	80
C.	Jou discloses a visualization system for generating interactive charts.	81
D.	Clark discloses a system for publishing charts to an enterprise website.....	85
E.	The QlikView Manuals disclose a visual analytics platform to create and publish interactive charts.....	86
F.	Bales discloses drag and drop for updating web content.	94
VIII.	Grounds of Unpatentability.....	99
IX.	Ground 1: Rostoker in view of Strong renders obvious claims 1-4.....	99
A.	Independent Claim 1	99
B.	Dependent Claim 2	149
C.	Dependent Claim 3	149
D.	Dependent Claim 4	149
X.	Ground 2: Rostoker and Strong in view of Jou renders obvious claims 1-4.	150
XI.	Ground 3: Rostoker and Strong in view of Clark renders obvious claims 1-4.	154
XII.	Ground 4: The QlikView Manuals in view of Bales render obvious claims 1, 3, and 4.	159
A.	A POSA would have been motivated to combine the QlikView Manuals.	159
B.	A POSA would have been motivated to combine the QlikView Manuals and Bales.	161

C. Independent Claim 1	165
D. Dependent Claim 3	205
E. Dependent Claim 4	206
XIII. No Objective Indicia of Nonobviousness	208
XIV. Conclusion.....	210

I. Introduction

I, Christopher Schmandt, declare as follows:

1. I have been retained on behalf of Salesforce, Inc. for the above-captioned *inter partes* review proceeding (IPR). I understand that this proceeding involves U.S. Patent No. 8,520,000 (EX1001, “the ’000 patent”), which is titled “Creation, Sharing and Embedding of Interactive Charts.” I am being compensated for my time in connection with this IPR at a rate of \$550 per hour. My compensation is not contingent on either my expert opinion or the outcome of this litigation. I have no other interest in this proceeding.

2. I understand the ’000 patent is currently assigned to iCharts LLC. I also understand that the ’000 patent names two inventors: Seymour Duncker and Tyron Montgomery. EX1001, (75).

3. I reviewed and am familiar with the specification and the claims of the ’000 patent. The ’000 patent was filed on February 17, 2009 and claims priority as a continuation-in-part of U.S. Patent Application No. 12/205,802 (“the ’802 application”), which was filed on September 5, 2008 and now U.S. Patent No. 8,271,892. EX1001, (22), (63). I understand the ’000 patent also claims priority to Provisional Application No. 61/077,728, which was filed on July 2, 2008. EX1001, (60).

4. I understand the ’000 patent has been provided as EX1001. I will cite

to the specification using the following format: EX1001, 1:1-10. This citation points to the specification of the '000 patent at column 1, lines 1-10.

5. I have reviewed and am familiar with the file history of the '000 patent. The file history of the '000 patent has been provided as EX1002.

6. I have reviewed and am familiar with the following prior art used in the petition for *inter partes* review of the '000 patent:

- EX1005: U.S. Publication No. 2008/0127052 to Rostoker;
- EX1006: U.S. Publication No. 2007/0113194 to Bales *et al.*;
- EX1007: U.S. Publication No. 2003/0071814 to Jou *et al.*;
- EX1008: QlikView Reference Manual V8.50 (June 2008);
- EX1009: QlikView Server Reference Manual V8.50 (June 2008);
- EX1010: U.S. Publication No. 2006/0112123 to Clark *et al.*; and
- EX1067: U.S. Publication No. 2006/0101064 to Strong *et al.*

7. I have been asked to provide my technical review, analysis, insights, and opinions about the '000 patent and the above-listed references that form the basis for the grounds of unpatentability. I reserve the right to supplement this declaration in response to additional evidence that may come to light.

II. List of Documents Considered

8. In forming my opinions, I have considered the documents listed below in addition to any other documents cited in this declaration. To the best of my

knowledge, these documents are true and accurate copies of what they purport to be. An expert in the field would reasonably rely on them to formulate opinions such as those set forth in this declaration.

Exhibit	Description
1001	U.S. Patent No. 8,520,000 to Duncker <i>et al.</i>
1002	U.S. Patent Prosecution History of U.S. Patent No. 8,520,000
1005	U.S. Publication No. 2008/0127052 to Rostoker (“Rostoker”)
1006	U.S. Publication No. 2007/0113194 to Bales <i>et al.</i> (“Bales”)
1007	U.S. Publication No. 2003/0071814 to Jou <i>et al.</i> (“Jou”)
1008	QlikView Reference Manual V8.50 (June 2008)
1009	QlikView Server Reference Manual V8.50 (June 2008)
1010	U.S. Publication No. 2006/0112123 to Clark <i>et al.</i> (“Clark”)
1011	Timothy J. O’Leary & Linda L. O’Leary, <i>Microsoft Office Excel 2003</i> (2003)
1012	Timothy J. O’Leary & Linda L. O’Leary, <i>Microsoft Office PowerPoint 2003</i> (2003)
1013	U.S. Patent Application No. 61/077,728

1014	QlikView Downloads (Aug. 2, 2008), <i>available at</i> https://web.archive.org/web/20080802064135/http://ftp.qliktech.com:80/qvwebdownloads/
1015	QlikView Product Suite Download Page (July 8, 2008), <i>available at</i> https://web.archive.org/web/20080708191653/http://ftp.qliktech.com/qvwebdownloads/Release8.asp
1016	QlikView 8.50 Downloadable Files (Aug. 17, 2008), <i>available at</i> https://web.archive.org/web/20080817042640/http://ftp.qliktech.com/qvwebdownloads/Release850.asp
1017	QlikView 8.50 Documentation (Aug. 17, 2008), <i>available at</i> https://web.archive.org/web/20080817042629/http://ftp.qliktech.com:80/qvwebdownloads/Docs.asp
1018	QlikView 8.50 Documentation (Oct. 23, 2008), <i>available at</i> https://web.archive.org/web/20081023184538/http://ftp.qliktech.com:80/qvwebdownloads/Docs.asp
1019	QlikView 8.50 Documentation (Feb. 9, 2009), <i>available at</i> https://web.archive.org/web/20090209142916/http://ftp.qliktech.com:80/qvwebdownloads/Docs.asp
1020	QlikView Community Support (July 2, 2008), <i>available at</i> https://web.archive.org/web/20080702053130/http://www.qlikcommunity.com/support/
1021	QlikNews: Issue 5, Summer 2008 (Mar. 4, 2009), <i>available at</i> https://web.archive.org/web/20090304004927/http://www.qlikcommunity.com/1203/

1022	QlikNews: ProductView (June 26, 2008), <i>available at</i> https://web.archive.org/web/20080626190929/http://www.qlikcommunity.com/1218/
1023	QlikNews: Americas View (June 26, 2008), <i>available at</i> https://web.archive.org/web/20080626195546/http://www.qlikcommunity.com/1213/
1024	QlikView Wine Guide Intro Tab (Jan. 7, 2007), <i>available at</i> http://web.archive.org/web/20070107033756/http://thewinepress2.qliktech.com/en/Default.asp
1025	QlikView Wine Guide How To Tab (Jan. 8, 2007), <i>available at</i> http://web.archive.org/web/20070108145241/http://thewinepress2.qliktech.com/en/HowTo.asp
1026	QlikView Wine Guide Wine Selector Tab (Jan. 7, 2007), <i>available at</i> http://web.archive.org/web/20070107034057/http://thewinepress2.qliktech.com/en/Wine_Selector.asp
1027	QlikView Wine Guide Analysis Tab (Jan. 8, 2007), <i>available at</i> http://web.archive.org/web/20070108140845/http://thewinepress2.qliktech.com/en/Analysis.asp
1028	James Richardson, <i>et al.</i> , <i>Magic Quadrant for Business Intelligence Platforms</i> , 2008, Gartner ID No. G00154227 (Feb. 1, 2008)
1029	David M. Raab, <i>Analytical Database Options</i> , DM Review (July 2008)
1030	Declaration of Sylvia D. Hall-Ellis, Ph.D.
1031	Curriculum Vitae of Sylvia D. Hall-Ellis, Ph.D.

1032	Internet Archive Declaration #1
1033	Internet Archive Declaration #2
1034	Redline Comparison – U.S. Application No. 12/205,802 Compared with U.S. Application No. 12/372,696 to Show New Matter
1035	Graham J. Wills, NicheWorks—Interactive Visualization of Very Large Graphs (1999)
1036	Jef Raskin, The Humane Interface (2000)
1037	Los Angeles Times, Jef Raskin, 61; Led the Team That Created User-Friendly Apple Macintosh Computer (Mar. 1, 2005)
1038	Ian Austen, I.B.M. Acquires Cognos, Maker of Business Software, for \$4.9 Billion, N.Y. Times, (Nov. 13, 2007)
1039	Viegas et al., ManyEyes: A site for visualization at internet scale, IEEE Transactions on Visualization and Computer Graphics 13(6), 1121-1128. (Jan. 1, 2007)
1040	Chris Stolte and Pat Hanrahan, Polaris: A System for Query, Analysis and Visualization of Multi-dimensional Relational Databases, IEEE Transactions on Visualization and Computer Graphics, Vol. 8, No. 1 (Aug. 7, 2002)
1041	Tableau Software, Visual Analysis for Everyone (2007)
1042	Jeremy Allaire, Macromedia Flash MX—A next-generation rich client (March 2002)

1043	Macromedia Flash (SWF) File Format Specification, v6 (2002)
1044	David Flanagan, JavaScript The Definitive Guide (4th ed. 2002)
1045	U.S. Patent No. 5,414,809 to Hogan et al.
1046	U.S. Patent No. 7,296,025 to Kung et al.
1047	U.S. Patent No. 6,025,844 to Parsons
1048	U.S. Patent No. 8,099,674 to Mackinlay et al.
1049	Tableau Overview Video, available at https://web.archive.org/web/20080124114801/ http://www.tableausoftware.com:80/products/tour
1050	U.S. Patent No. 5,848,424 to Scheinkman et al.
1051	U.S. Patent Publication No. 2007/0243925 to LeMay et al.
1052	U.S. Patent Publication No. 2005/0193330 to Peters
1053	Dance et al., Faces of the Dead, The New York Times, https://archive.nytimes.com/www.nytimes.com/interactive/us/faces- of-the-dead.html (last visited Aug. 7, 2024)
1054	U.S. Patent No. 5,535,324 to Alvarez et al.
1055	U.S. Patent No. 6,529,217 to Maguire et al.
1056	George Shepherd and Scot Wingo, MFC Internals, (1996)

1057	Tom Negrino and Dori Smith, JavaScript and Ajax (2007)
1058	Plaintiff iChart’s First Amended Asserted Claims, Infringement Contentions, and Document Production Disclosures (Patent Local Rules 3-1, 3-2), Case No. 3:24-cv-03157 (with Exhibit C)
1059	QlikView 8.50 Copyright Registration
1060	QlikView 8.50 Copyright Application
1061	Qlik Community Page, Qlikview 8.5 Manual, available at https://community.qlik.com/t5/QlikView-Administration/Qlikview-8-5-Manual/m-p/1742630
1064	Nikolaïdou <i>et al.</i> “Enterprise Information Systems Configuration: Emphasizing the Symbiotic Relationship Between Applications and the Underlying Network,” 9th IEEE International Conference on Engineering Complex Computer Systems Navigating Complexity in the e-Engineering Age (2004)
1065	Widgets 1.0: The Widget Landscape (Q1 2008), W3C
1066	Microsoft Computer Dictionary 5th Edition (2002)
1067	U.S. Publication No. 2006/0101064 to Strong et al.
1068	Declaration of Deborah C. Lofton, Chief Legal Officer and Corporate Secretary of QlikTech Inc.

III. Qualifications

9. I retired several years ago after a 40-year career at the Massachusetts Institute of Technology (“MIT”). For most of that time, I was employed as a

Principal Research Scientist at the Media Laboratory, of which I was a co-founder.

I also served as faculty for the MIT Media Arts and Sciences academic program.

10. I received my Bachelor of Science degree in Electrical Engineering and Computer Science from MIT in 1978, and my Master of Science degree in Visual Studies (Computer Graphics) from MIT in 1980.

11. I was employed at MIT from 1980 to 2018, initially at the Architecture Machine Group, an early computer graphics research Lab. In 1985, I helped found the Media Laboratory and continued to work there until retirement.

12. I was also Director of the Living Mobile Research Group. My research related to the enhancement of personal communication and virtual social awareness, as well as conversational user interfaces. As part of my work, I developed novel applications and user interfaces. I also worked on distributed and online communication and collaborative systems, with an emphasis on multi-media (including images and video) and user interfaces.

13. In my faculty position, I taught courses and supervised student research and theses at the Bachelors, Masters, and Ph.D. levels. I also oversaw the Masters and Ph.D. thesis programs for the entire MIT Media Arts and Sciences academic program.

14. I have more than 70 published conference and journal papers and one book in these fields. A list of my published conference and journal papers are included in my *Curriculum Vitae*, which has been submitted as Exhibit 1004.

15. In the course of my education and research career, I had early opportunities to build graphical user interfaces employing a variety of interaction techniques for the purpose of visualizing different data. For example, my undergraduate thesis in 1978 was one of the earliest examples of an electronic book, which used multiple display screens, menus for navigation, multiple gesture-based input devices, and even employed animation to simulate page turning.

16. In the early 1980s, I built pioneering systems to visualize and edit sound files using a waveform-like representation, and to manage real time streams of video arranged on a “wall” and activated by either touch or eye tracking. In the early 1990s, I built a number of personal information systems, including calendar, rolodex, telephone dialer, and email reader systems. These personal information systems ran under the X Window System with the goal, in part, to interoperate these applications, including use of drag-and-drop techniques. For example, a user could select a phone number from the rolodex and cause the computer to dial it (using an attached speaker phone) by using the X Window conventions to select the number from the rolodex and drag it to the dialer. Similarly, a user could select a portion of a voice message from a graphic display and drag-and-drop it into a

calendar. These applications typically utilized multiple graphic “panes” to separate application data from the user interface elements such as menus, as was well known and practiced at the time.

17. In the period from 1983-1987, I co-founded a startup, Trionum, with one other programmer to build a structured graphics editor, which we named Pep. At the time, paint programs were well known, but the “paint” was simply pixels. A structured editor allowed a user to create graphic objects, which could then be moved, resized, duplicated, rotated, etc. When one of these objects was moved, it would show drag-and-drop behavior, with the whole object being attached to the mouse cursor and moving in real time to its new desired position.

18. Pep also included an add-on that provided limited charting, in the form of bar and pie graphs. Pep could import Lotus 1-2-3 charts and “decompose” them for editing, such as changing the color or order of bars or pulling out a piece of the pie for emphasis. Although this product was only sold in small quantities to hobbyists, Lotus purchased the 1-2-3 interface code to be included in their Symphony product line.

IV. Legal principles

19. I am not a lawyer. However, I have been advised that technical experts should apply certain legal standards in forming their opinions.

A. Claim Construction

20. I understand that claim terms should be interpreted under a claim construction standard called *Phillips*. I understand that, according to the *Phillips* standard, claim terms should be construed according to the terms' meaning as understood by a person having ordinary skill in the art (POSA) at the time of the invention. I understand that the claim terms should be construed in light of the specification. I also understand that the prosecution history of the patent should be consulted in determining claim meaning. In this declaration, I have construed the terms according to their ordinary and customary meanings as understood by a POSA in view of the specification and prosecution history.

B. Obviousness

21. I understand a patent claim is unpatentable if the claimed invention would have been obvious to a person of ordinary skill in the art (POSA) at the time of the alleged invention. This means that a patent claim can be unpatentable even if all of the claim elements cannot be found in a single prior-art reference.

22. I understand an invention is obvious when the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious to a POSA at the time of the invention.

23. I understand that to prove that prior art renders a patent claim obvious, it is necessary to: (1) identify the particular references that, singly or in

combination, teach or suggest every element of the claim; (2) identify which claim elements appear in which reference; and (3) explain why a POSA would have sought to combine the prior-art references.

24. I understand that when considering whether a patent claim is obvious, one should consider whether at the time of the alleged invention a teaching, suggestion, or motivation to combine the references existed so as to avoid impermissibly applying hindsight. I understand this test should not be rigidly applied, but that the test can be important to avoid such hindsight.

25. I understand the following rationales can support a motivation to combine prior-art references:

- (A) Combining prior art elements according to known methods to yield predictable results;
- (B) Simple substitution of one known element for another to obtain predictable results;
- (C) Use of known technique to improve similar devices (methods, or products) in the same way;
- (D) Applying a known technique to a known device (method, or product) ready for improvement to yield predictable results;
- (E) Choosing from a finite number of identified, predictable solutions, with a reasonable expectation of success; and

(F) Known work in one field of endeavor may prompt variations of it for use in either the same field or a different one based on design incentives or other market forces if the variations are predictable to one of ordinary skill in the art.

26. I understand objective indicia (also called secondary considerations) can be important evidence as to whether a patent claim is obvious. Such indicia include: (1) commercial success of products covered by the patent claim; (2) a long-felt need for the invention; (3) failed attempts by others to make the invention; (4) copying of the invention by others; (5) unexpected results achieved by the invention; (6) praise of the invention by the infringer or others in the field; (7) taking of licenses under the patent by others; (8) expressions of surprise by experts and those skilled in the art at the making of the invention; and (9) the patentee proceeded contrary to the accepted wisdom of the prior art.

C. Written Description

27. I understand that a patent specification shall contain a written description of the claimed invention.

28. I understand the written description requirement has several policy objectives: (1) to satisfy the inventor's obligation to disclose the technologic knowledge upon which the patent is based and (2) to demonstrate that the inventor was in possession of the invention that is claimed. To satisfy the written description requirement, I understand a patent specification must describe the

claimed invention in sufficient detail that one skilled in the art can reasonably conclude that the inventor had possession of the claimed invention.

29. I understand that adding new matter into a claim during prosecution can give rise to a written description issue. Particularly, I understand that new or amended claims that introduce new claim limitations that are unsupported by the as-filed disclosure violate the written description requirement.

D. A Person of Ordinary Skill in the Art

30. I understand that a POSA is presumed to be aware of all pertinent art, thinks along conventional wisdom in the art, and is a person of ordinary creativity—not an automaton.

31. I have been asked to consider the level of ordinary skill in the field that someone would have had at the time the claimed invention was made. In deciding the level of ordinary skill, I considered the following:

- the levels of education and experience of persons working in the field;
- the types of problems encountered in the field; and
- the sophistication of the technology.

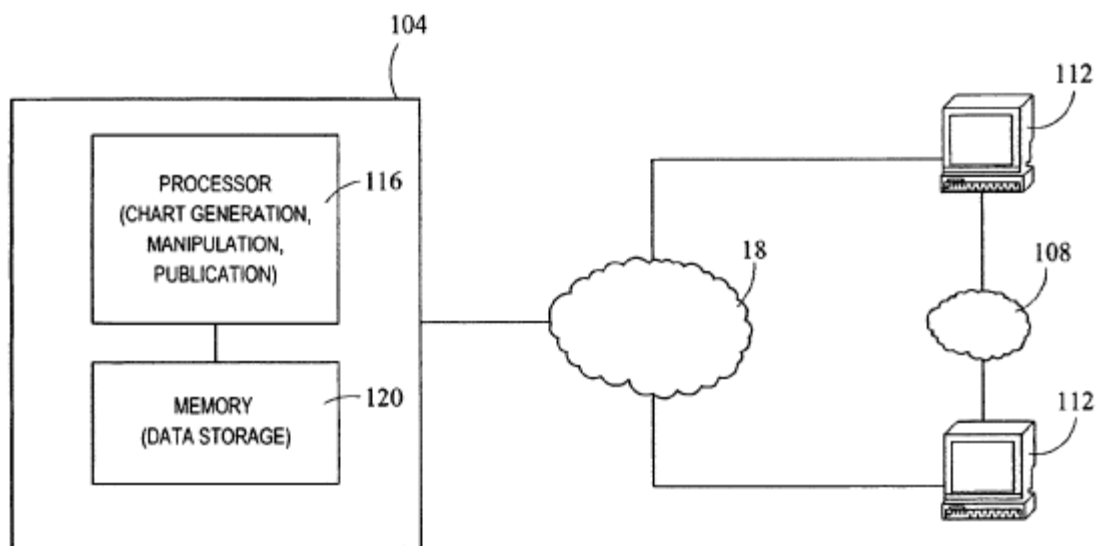
32. My opinion below explains how a POSA would have understood the technology described in the references I have identified herein around the February 17, 2009 timeframe. However, my opinions herein would not be affected if the '000 patent is found to be entitled to an earlier priority date.

33. Regardless of whether I use “I” or a “POSA” during my technical analysis below, all of my statements and opinions are always to be understood to be based on how a POSA would have understood or read a document at the time of the alleged invention.

V. U.S. Patent No. 8,520,000

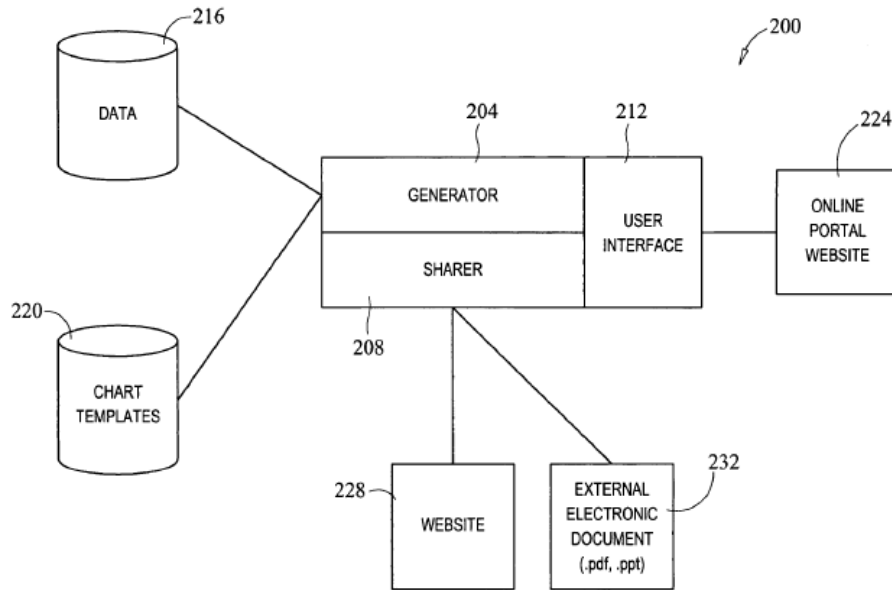
34. The ’000 patent describes “[s]ystems and methods for generating and sharing interactive charts.” EX1001, Abstract.

35. The ’000 patent generates and shares interactive charts using a server and client computers. As shown below, “server 104 is configured to receive requests from computers 112, process the requests and provide responses to the requests to the computers 112.” EX1001, 6:50-52. “In one embodiment, the server 104 generates an interactive chart based on input received from one of the computers 112 and shares the interactive chart with users located at one of the computers 112 over the network 108.” EX1001, 7:17-23.



EX1001, FIG. 1.

36. Figure 2 (below) shows “interactive chart online portal 200” EX1001, 7:45-47. “The interactive chart online portal 200 includes an interactive chart generator 204, an interactive chart sharer 208, a user interface 212, a data store 216 and a chart template data store 220.” EX1001, 7:53-56. “The user interface 212 is accessed through a website 224 which provides an online portal to the interactive chart generator 204 and the interactive chart sharer 208.” EX1001, 7:62-64. “Users upload or enter data at the website 224 through the user interface 212 which stores the data in the data store 216.” EX1001, 8:1-3. “The interactive chart generator 204 uses the data and chart templates stored in the chart template data store 220 to generate the interactive charts.” EX1001, 8:3-5.



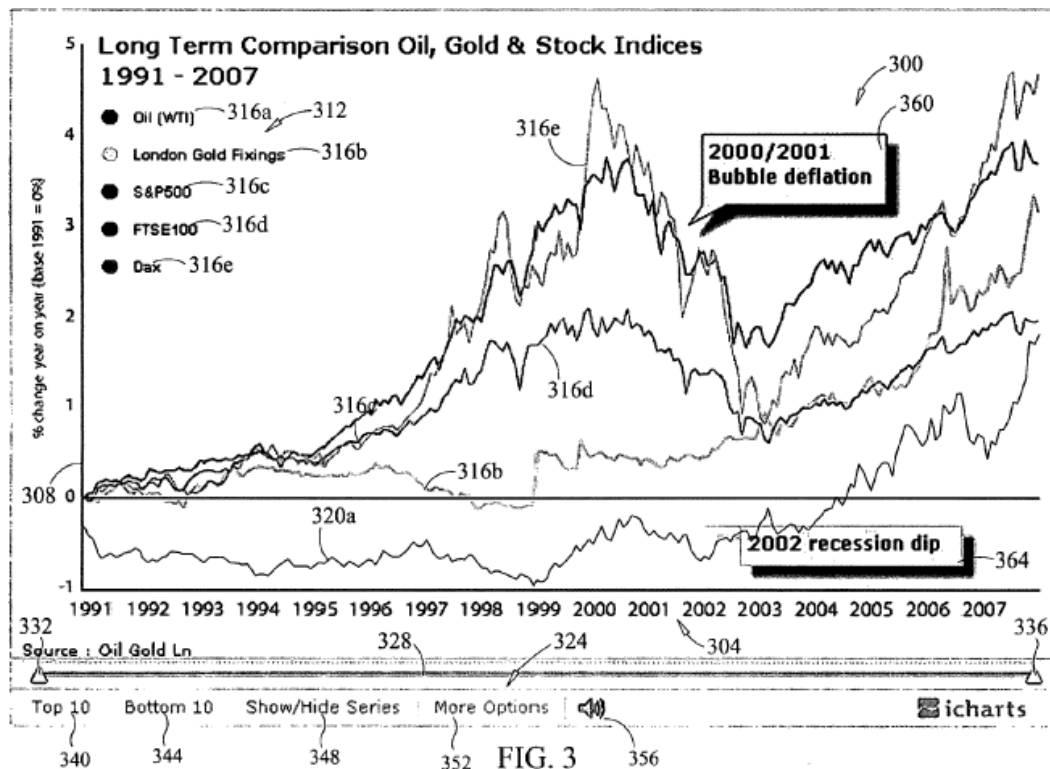
EX1001, FIG. 2.

37. Shown above, “[t]he generated interactive chart can be published on an external webpage 228 as an interactive chart or a static chart.” EX1001, 8:7-9. “The generated interactive chart can also be embedded in an external document 232 as an interactive chart or a static chart.” EX1001, 8:9-11. The ’000 patent describes external electronic documents broadly as including Microsoft Excel files, Microsoft Word files, web pages, portable document files (PDFs), PowerPoint files, etc. *See* EX1001, 1:23-26, 8:11-14, 13:30-33.

38. In one embodiment, “[t]he interactive charts that are downloaded are self-contained flash files that can exist independent of the web portal.” EX1001, 8:47-48. In this embodiment, “the interactive chart 300 is a flash file that contains code that a) renders the chart, b) obtains input from the user as per his desired

action on this chart (i.e., interactivity), and c) rerenders the chart using the input as parameters.” EX1001, 8:23-27.

39. The '000 patent refers to “interactive” charts broadly to include different interactive features. For example, shown below, “chart 300 has interactive features that allow a user to interact with the chart to, for example, modify or change their view of the chart, learn more about the chart data, and the like.” EX1001, 9:9-13. “Exemplary interactive features include a zoom feature, a data mouse-over pop-up window, a data hide/share feature, an audio playback feature, a drill down feature, a data gliding feature and the like.” EX1001, 9:13-16.



EX1001, FIG. 3.

40. The '000 patent describes an embodiment that provides a method that includes “providing a tool on a first website for embedding at least one interactive chart in a chart box, the chart box embedded on a second website,” and “allowing a user to select at least one of the plurality of interactive charts to be embedded in the chart box, wherein the at least one of the plurality of interactive charts is automatically displayed in the chart box on the second website when the user selects the at least one of the plurality of interactive charts” in the first website. EX1001, 2:10-22. The user selects interactive charts “by dragging and dropping the at least one of the plurality of interactive charts into a chart box folder of the tool” in the first website. EX1001, 2:23-26.

41. Figure 43 below “illustrates an exemplary screen shot of a business services application website 4300.” EX1001, 15:49-51. “The illustrated website 4300 includes a chart box 4304,” and “the chart box 4304 is configured to be embedded on a website to display one or more interactive charts on that website.” EX1001, 15:51-55. “The chart(s) are added to the chart box using a chart box tool provided by another website that allows users to generate, purchase or otherwise manipulate and select the interactive chart(s).” EX1001, 15:55-58.

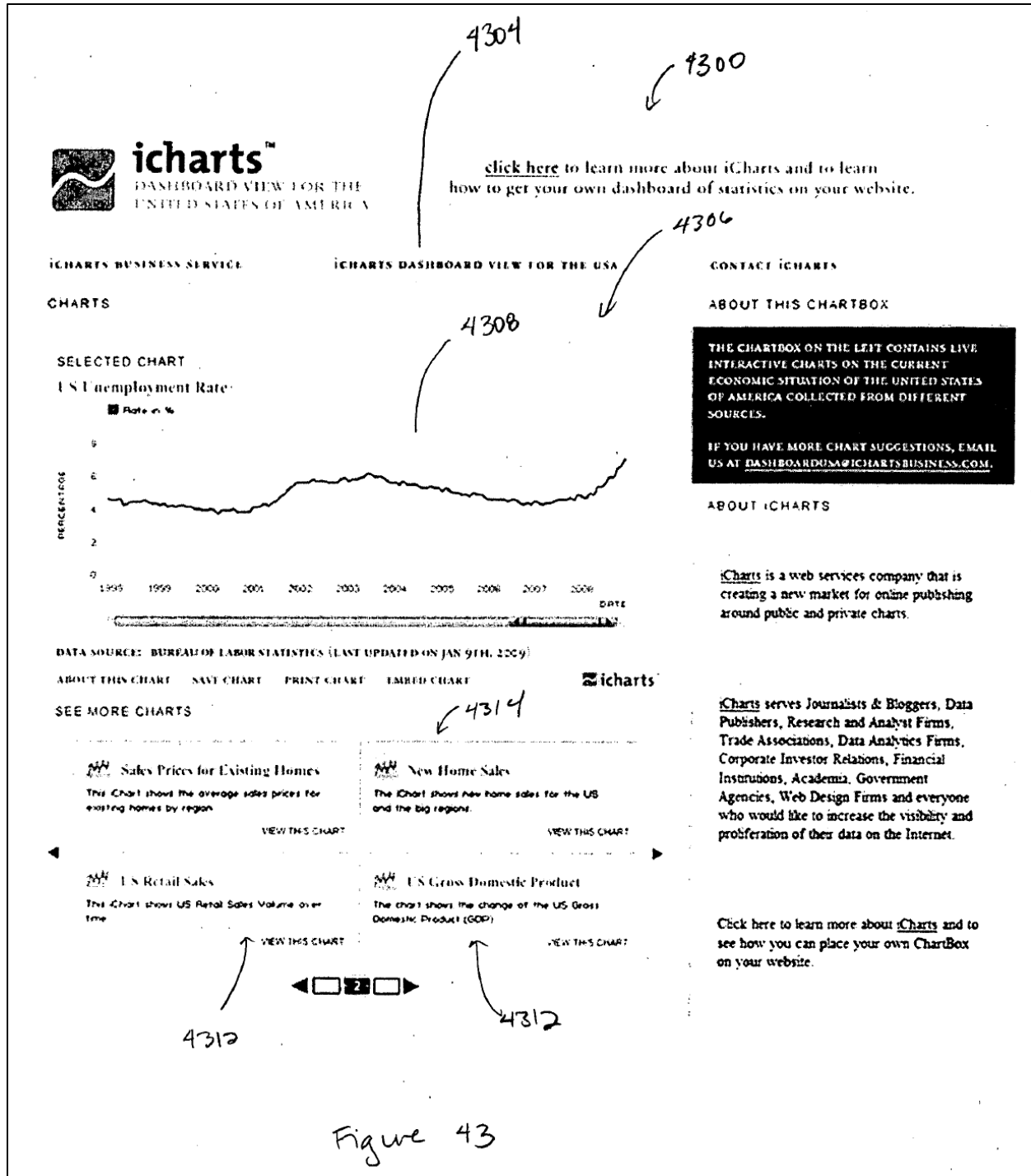


Figure 43

EX1001, FIG. 43.

42. "In one embodiment, the chart box 4304 is a flash widget." EX1001, 16:1. As illustrated in Figure 43, "[t]he [] chart box 4304 includes a viewing pane

4306 configured to display an interactive chart 4308.” EX1001, 16:11-12. “The chart box 4304 also includes multiple interactive charts 431 in a chart selection pane 4314.” EX1001, 16:12-14.

43. “As shown in FIG. 45, a user interface 4500 may be provided . . . on another website coupled a server that includes several interactive charts that can be selected for display in the chart box 4304.” EX1001, 16:25-29. “The user interface also includes a chart box region 4516 that includes a chart box folder 4520,” which “is connected to the chart box 4304 such that the retrieve code for data feed displays the interactive chart(s) selected by the user in the chart box folder 4520.” EX1001, 16:33-38. “The user can select the interactive chart(s) . . . by, for example, dragging and dropping one or more of the charts 4512 onto the chart box folder 4520.” EX1001, 16:38-41. Then “the chart is automatically displayed in the chart box 4304 embedded on the user’s website.” EX1001, 16:41-44.

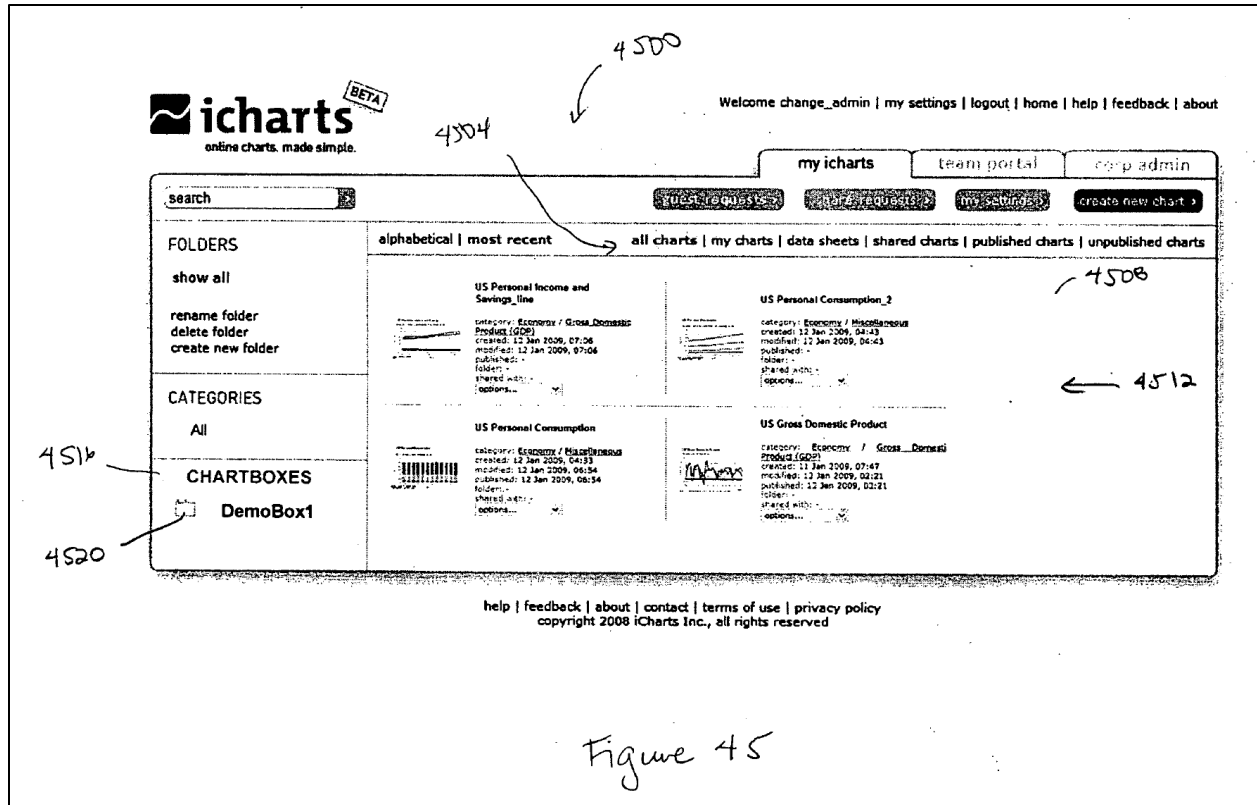


Figure 45

44. The '000 patent also describes an embodiment that provides “an online portal or user interface” that “allows users to drag and drop the data from a data pane to a charting pane, automatically creating the interactive chart.” EX1001, 5:13-15. In one embodiment shown below, “[w]hen a user clicks on the charting pane 1400, several chart types 1500 are displayed in the chart type tab 1424 of the data pane 1404, as shown in FIG. 15.” EX1001, 11:30-32.

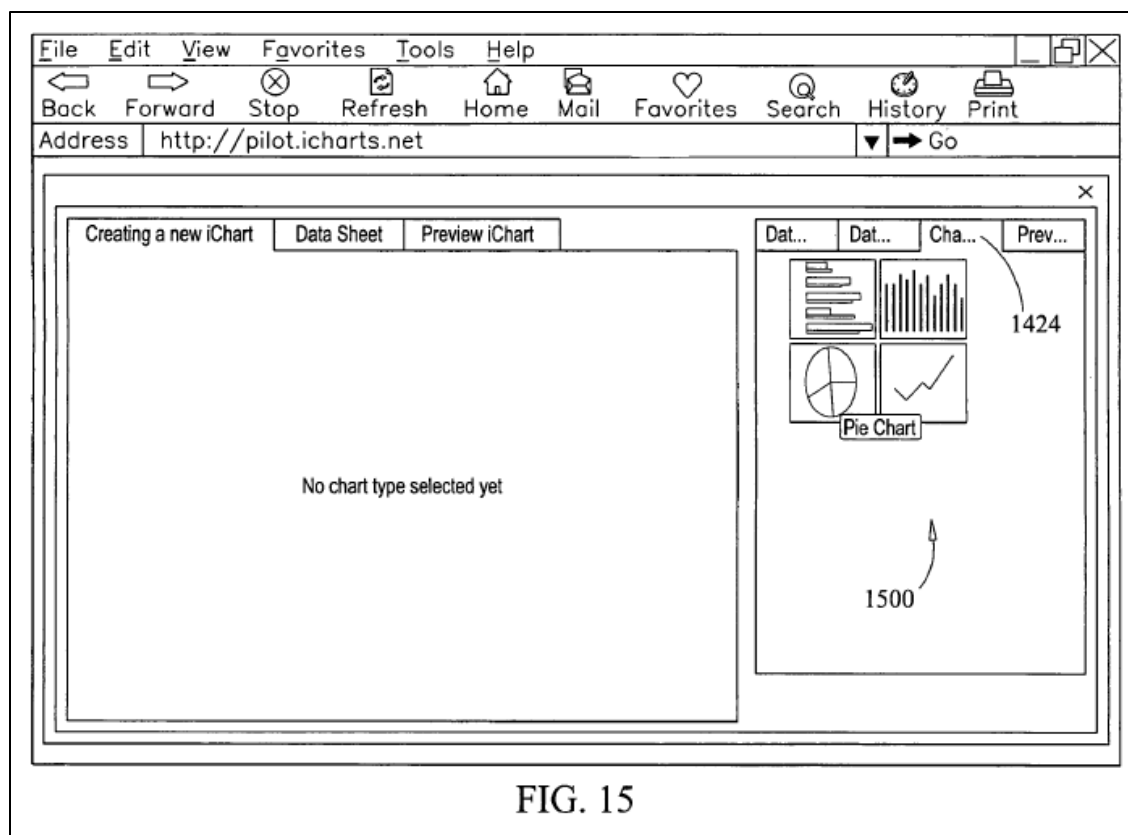
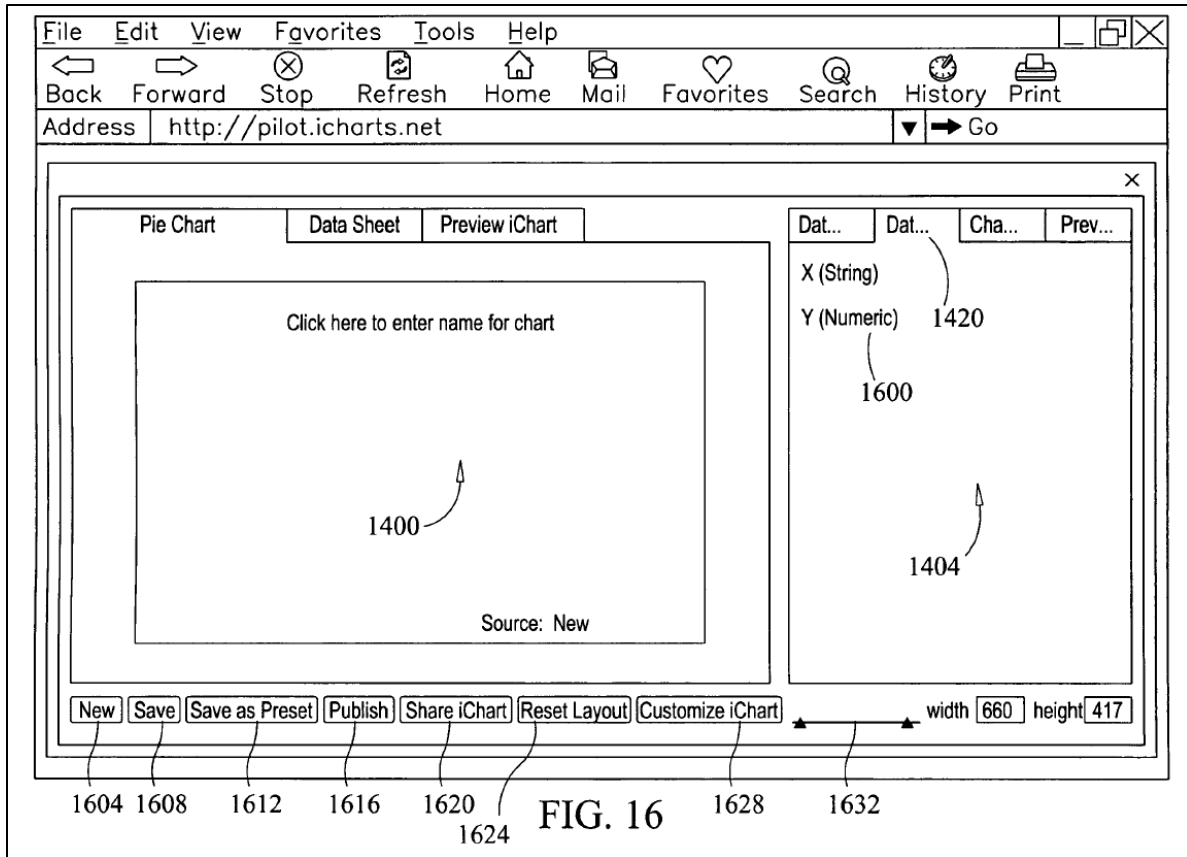


FIG. 15

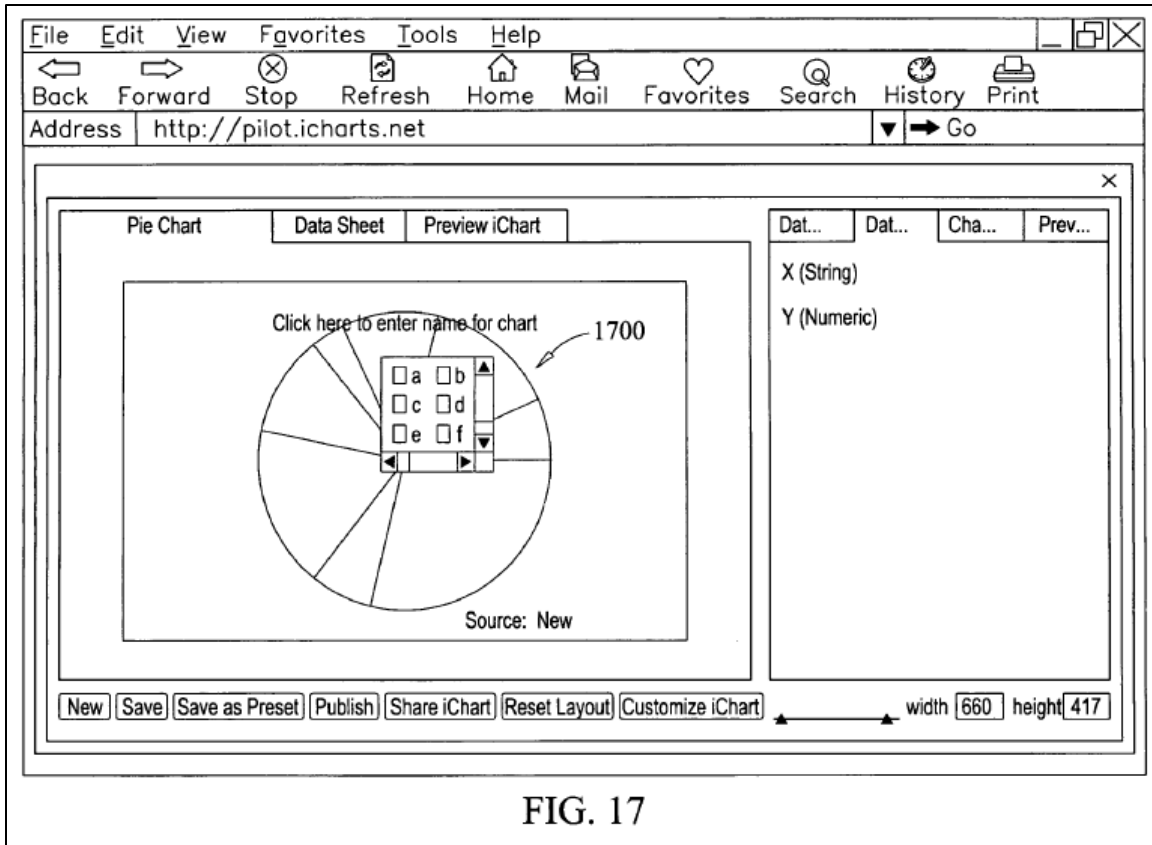
EX1001, FIG. 15.

45. “When a user selects a chart type 1500, a blank charting area is shown in the charting pane 1400, as shown in FIG. 16 [(shown below)].” EX1001, 11:39-40. “The user then selects data from the data pane 1404 in the detailed data tab 1420 by clicking on the data with the mouse.” EX1001, 11:40-41. “The user continues to click on the data and drags and drops the data from the data pane 1404 and onto the chart pane 1400” EX1001, 11:42-44.



EX1001, FIG. 16.

46. “When the user drags and drops the data on the chart pane 1400, an interactive chart 1700 is automatically generated as shown in FIG. 17 [(shown below)].” EX1001, 11:56-58. “The user can then manipulate the chart 1700 to have the look and feel and interactive features that the user desires” EX1001, 12:3-5.



EX1001, FIG. 17.

A. Claims

47. The '000 patent issued with 4 claims, one of which is independent.

Independent claim 1 is shown below. As I show below, the claim recites a method that “select[s], on a first website, . . . at least one interactive chart” and “display[s] . . . the at least one interactive chart . . . on a second website,” “wherein the selection is made by dragging and dropping the at least one interactive chart from the plurality of charts into a chart box folder connected to the chart box.”

Independent Claim 1

A computer-implemented method comprising:

receiving a selection, on a first website, of at least one interactive chart from a plurality of interactive charts; and

displaying, in response to the selection the at least one interactive chart embedded in a chart box on a second website, wherein the chart box is a web widget,

wherein the at least one interactive chart is generated on a generation interface configured to allow a user to generate the interactive chart with a chart template from a chart template store,

wherein the at least one interactive chart, when displayed on the second website in the absence of the generation interface, enables a user to a) render the chart, b) obtain input from a user and c) rerender the chart using the input thereby enabling the interactive chart to retain interactivity,

wherein the selection is made by dragging and dropping the at least one interactive chart from the plurality of charts into a chart box folder connected to the chart box such that retrieve code for data feed displays the at least one interactive chart on the first website, and

wherein the at least one interactive chart is automatically displayed on the second website using a data feed from the first website to the second website upon the dragging and dropping selection.

EX1001, 18:23-49.

B. Prosecution History of the '000 Patent

48. The '000 patent issued from U.S. Patent Application No. 12/372,696 (“the '696 application”), which was filed on February 17, 2009. EX1001, (21), (22).

49. On April 1, 2011, the Office issued a Non-Final Office Action. EX1002, 0127-141. The Office objected to figures (also called drawings) “because they include the following reference character(s) not mentioned in the description[.]” EX1002, 0129. The Office also objected to the specification because of informalities. EX1002, 0130. Finally, the Office rejected all pending claims (1-20) based on the following grounds of rejection:

Claims 1-6 are rejected under 35 U.S.C. 103(a) as being unpatentable over Waldorf (U.S. Patent 7,667,582; hereinafter Waldorf) in view of Chiluvuri (U.S. Patent 7,840,937; hereinafter Chiluvuri).

EX1002, 0131.

Claims 7-8, 10-13, 15-17 and 19-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over Waldorf (U.S. Patent 7,667,582; hereinafter Waldorf) in view of Louw (U.S. Publication 2007/0203945; hereinafter Louw).

EX1002, 0134.

Claims 8, 14 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Waldorf as modified by Louw as applied to claims 13, 15-17 and 19-20 above, and further in view of Chiluvuri (U.S. Patent 7,840,937; hereinafter Chiluvuri).

EX1002, 0138.

50. The Office found that Waldorf teaches a method of “sending a request from a first server (client) to a second server (server) to retrieve the data for an interactive chart; receiving the data of the interactive chart from the second server at the first server and displaying the interactive chart at the website hosted by second server.” EX1002, 0131. The Office also found that Waldorf teaches the “request is sent via chart box, embedded in a webpage and also displayed as chart box embedded in webpage.” EX1002, 0132. The Office relied on Chiluvuri for rendering charts “as flash files.” EX1002, 0132-33. The Office relied on Louw for “dropping the chart into a chart folder.” EX1002, 0134-35.

51. In response to the objections and rejections, the Applicant amended the figures and specification. The Applicant also amended the claims and added four new dependent claims (claims 21-24). EX1002, 0150-58. I have shown the amendment to the first independent claim (1) below where underlined text is new claim language and cross-through text is removed claim language.

1. (Currently Amended) A computer-implemented method comprising:
~~receiving sending a request from a first server to a second server to retrieve a flash file of an interactive chart;~~
~~receiving the flash file of the interactive chart from the second server at the first server;~~ and
displaying, in response to the request, the interactive chart ~~[[at]] embedded in a chart box on a website hosted by the second server, wherein the interactive chart retains interactivity when embedded.~~

EX1002, 0153.

52. Additionally, the Applicant amended independent claim (7) to recite steps of “receiving a selection” and “displaying” in response to the selection. Amended claim (7) also recited “wherein the at least one interactive chart retains interactivity when embedded”:

7. (Currently Amended) A computer-implemented method comprising:
receiving a selection, on a first website, of at least one interactive chart from a plurality of interactive charts; and
displaying, in response to the selection the at least one interactive chart embedded in a chart box on a second website, wherein the at least one interactive chart retains interactivity when embedded.
~~providing a tool on a first website for embedding at least one interactive chart in a chart box, the chart box embedded on a second website, the first website hosted by a first server that stores a plurality of interactive charts, the at least one interactive chart embedded in the chart box being one of the plurality of interactive charts stored on the first server; and~~
~~allowing a user to select at least one of the plurality of interactive charts to be embedded in the chart box, wherein the at least one of the plurality of interactive charts is automatically displayed in the chart box on the second website when the user selects the at least one of the plurality of interactive charts.~~

EX1002, 0153-54.

53. Furthermore, the Applicant amended independent claim (13) to recite steps of “receiving a user’s drag and drop” selection and “displaying” in response to the drag and drop selection. Amended claim (13) also recited “wherein the interactive chart retains interactivity when embedded”:

13. (Currently Amended) A computer-implemented method comprising:
receiving a user’s allowing a user to drag and drop, on a first website, of at least one of a plurality of interactive charts displayed in a user interface into a chart box folder of the user interface; and
displaying transmitting, in response to drag and drop, the at least one of the plurality of interactive charts ~~chart~~ from a first website to embedded in a chart box embedded in on a second website in response to the user dropping the at least one of the plurality of interactive charts into the chart box folder, wherein the interactive chart retains interactivity when embedded.

EX1002, 0155.

54. The Applicant amended independent claim (17) similarly.

17. (Currently Amended) A computer system comprising:
memory configured to store a plurality of interactive charts; and
a processor coupled to the memory, the processor configured to
receive a user’s drag and drop, on a first website, of at least one of a plurality of interactive charts into a chart box folder; and
display, in response to the drag and drop, the at least one interactive chart embedded in a chart box on a second website, wherein the interactive chart retains interactivity when embedded allow a user to drag and drop at least one of the plurality of interactive charts into a chart box pane of a user interface accessed at a first website hosted by a first server and transmit the at least one of the plurality of interactive charts from the first server to a second server for display of the at least one of the plurality of interactive charts on a second website hosted by the second server.

EX1002, 0155.

55. The Applicant argued that none of the cited references disclose an interactive chart as claimed. EX1002, 0160. According to the Applicant, the Office Action alleged that Waldorf and Louw taught customizable charts and reports. The Applicant argued that “customizability of a chart does not make the chart interactive.” EX1002, 0160. The Applicant provided two dictionary definitions to argue that customizability of a chart is not “interactive.” EX1002, 0160-61. The Office, however, did not adopt these definitions for the claimed term “interactive.” Instead, the Office raised new rejections based on different prior art. EX1002, 0189-200.

56. Specifically, on August 6, 2011, the Office issued a Final Office Action, rejecting all pending claims based on the following rejections:

Claims 1-3 and 5-6 are rejected under 35 U.S.C. 103(a) as being unpatentable over Waldorf (U.S. Patent 7,667,582; hereinafter “Waldorf”) in view of Tolle et al. (U.S. Patent 7,961,188; hereinafter “Tolle”).
EX1002, 0192.

Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Waldorf as modified by Tolle as applied to claims 1-3 and 5-6 above, and in view of Seven et al. (U.S. Publication 2009/0018961; hereinafter “Seven”).
EX1002, 0194.

Claims 7, 8, 10, 11, 13, 15-17 and 19-20 [are] rejected under 35 U.S.C. 103(a) as being unpatentable over Waldorf as modified by Tolle as applied to claims 1 above, in view of Louw (U.S. Publication 2007/0203945; hereinafter Louw).

EX1002, 0195.

Claims 9, 14 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Waldorf as modified by Tolle in view of Louw as applied to claims 7, 8, 10, 11, 13, 15-17 and 19-20 above, and further in view of Seven et al. (U.S. Publication 2009/0018961; hereinafter “Seven”).

EX1002, 0197.

57. On December 16, 2011, the Applicant filed a Request for Continued Examination (RCE). EX1002, 0217. The request amended independent claims to add the underlined text as follows:

Claim 1

1. (Currently Amended) A computer-implemented method comprising:
receiving a request to retrieve an interactive chart; and
displaying, in response to the request, the interactive chart embedded in a chart box on a website,
wherein the interactive chart is generated by a chart generator on a server with a chart template from a chart template store on a server, and retains interactivity when embedded as a self-contained electronic document independent of the chart generator.

EX1002, 0222.

Claim 7

7. (Currently Amended) A computer-implemented method comprising:
- receiving a selection, on a first website, of at least one interactive chart from a plurality of interactive charts; and
 - displaying, in response to the selection the at least one interactive chart embedded in a chart box on a second website,
- wherein the at least one interactive chart is generated by a chart generator on a server with a chart template from a chart template store on a server, and retains interactivity when embedded as a self-contained electronic document independent of the chart generator.

EX1002, 0222-23.

Claim 13

13. (Currently Amended) A computer-implemented method comprising:
- receiving a user's drag and drop, on a first website, of at least one of a plurality of interactive charts into a chart box folder and
 - displaying , in response to drag and drop, the at least one interactive chart embedded in a chart box on a second website , wherein the interactive chart is generated by a chart generator on a server with a chart template from a chart template store on a server, and retains interactivity when embedded as a self-contained electronic document independent of the chart generator.

EX1002, 0223.

Claim 17

17. (Currently Amended) A computer system comprising:
- memory configured to store a plurality of interactive charts; and
 - a processor coupled to the memory, the processor configured to
 - receive a user's drag and drop, on a first website, of at least one of a plurality of interactive charts into a chart box folder; and
 - display, in response to the drag and drop, the at least one interactive chart embedded in a chart box on a second website, wherein the interactive chart is generated by a chart generator on a server with a chart template from a chart template store on a server, and retains interactivity when embedded as a self-contained electronic document independent of the chart generator.

EX1002, 0224.

58. According to the Applicant, “[t]he cited references, even in combination, do not teach an interactive chart that is generated with a chart generator and template from a server and yet retains interactivity, when embedded as a self-contained document independent of the chart generator.” EX1002, 0227. For example, the Applicant argued:

First, as provided above, the interactive chart of the present claims takes advantage of the full power of a chart generator and chart templates located on a server. Nevertheless, when the interactive chart is embedded as a self-contained electronic document independent of the chart generator, the interactive chart is still interactive. Such a particular kind of interactive chart, however, is not disclosed in either Tolle or an embedded object/controls/app of the common knowledge, Applicants submit.

Since none of the art of record discloses or suggest an interactive chart the present claims prescribe, the Office fails to establish a prima facie case of obviousness.

EX1002, 0228-29.

59. The Office issued another Non-Final Office Action on February 14, 2012, again rejecting all claims based on the following grounds:

Claims 1-3 and 5-6 are rejected under 35 U.S.C. 103(a) as being unpatentable over Waldorf (U.S. Patent 7,667,582; hereinafter “Waldorf”) in view of Tolle et al. (U.S. Patent 7,961,188; hereinafter “Tolle”) and further in view of Tolle et al. (U.S. Publication 2006/0136819; hereinafter “Tolle819”).

EX1002, 0240, 0243, 0247.

In regard to dependent claim 21, Waldorf as modified by Tolle in view of Tolle819 teaches the interactive chart comprises code (obvious), when executed, render the chart, b) obtain input from a user and c) render the chart using the input as a parameter (Tolle, figure 8A-8B; “Drink being input parameter”).

EX1002, 0243.

Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Waldorf as modified by Tolle in view of Tolle819 as applied to claims 1-3 and 5-6 above, and in view of Seven et al. (U.S. Publication 2009/0018961; hereinafter “Seven”).

EX1002, 0243.

Claims 7, 8, 10, 11, 13, 15-17 and 19-20 [are] rejected under 35 U.S.C. 103(a) as being unpatentable over Waldorf as modified by Tolle in view of Tolle819 as applied to claim 1 above, in view of Louw (U.S. Publication 2007/0203945; hereinafter Louw).

EX1002, 0245.

In regard to dependent claims 22, 23 and 24, Waldorf as modified by Tolle in view of Tolle819 teaches the interactive chart comprises code (obvious), when executed, render the chart, b) obtain input from a user and c) render the chart using the input as a parameter (Tolle, figure 8A-8B; “Drink being input parameter”).

EX1002, 0247.

Claims 9, 14 and 18 are rejected under 35 U.S.C. 103(a) as being unpatentable over Waldorf as modified by Tolle in view of Tolle819 and Louw as applied to claims 7, 8, 10, 11, 13, 15-17 and 19-20 above, and further in view of Seven et al. (U.S. Publication 2009/0018961; hereinafter “Seven”).

EX1002, 0247.

60. After these new rejections, the Applicant initiated an examiner interview. The interview summary stated that the parties discussed a proposed amendment that appeared to put the application to prosecute further in good direction.

Discussed proposed amendment and cited art of record and said proposed amendment appears to put the application to procecute [sic] further in good direction.

EX1002, 0263.

61. On August 13, 2012, the Applicant made the following amendments to the independent claims. According to the Applicant, “the independent claims prescribe two levels of interactivity for the chart, (1) one that requires the presence

of the generation interface and can take advantage of the chart template store when generating the chart, and (2) the other that is independent of the generation interface.” EX1002, 0272 (emphasis in original).

Claim 1

1. (Currently Amended) A computer-implemented method comprising:
receiving, at a server, a request to retrieve an interactive chart; and
displaying in response to the request, the interactive chart ~~embedded in a chart box~~ on a website,
wherein the interactive chart is generated on a generation interface configured to allow a user to generate the interactive chart by a chart generator on a server with a chart template from a chart template store ~~on a server~~, and
wherein the interactive chart, when displayed on the website in the absence of the generation interface, enables a user to a) render the chart, b) obtain input from a user and c) rerender the chart using the input thereby enabling the interactive chart to retain ~~retains~~ interactivity ~~when embedded as a self-contained electronic document independent of the chart generator~~.

EX1002, 0266.

62. Independent claims 7, 13 and 17 were likewise amended. Claims 19 and 21-24 were canceled. EX1002, 0270. The Applicant argued that “[s]uch two-level interactivity, however, is not present in any art of record, or the combination thereof.” EX1002, 0272. According to the Applicant, the Office recognized that Waldorf and Tolle were silent on generating charts with a chart template. Tolle⁸¹⁹ taught the generation of charts using a template but did not teach that the chart generated with the template retained interactivity in the absence of the generation interface. EX1002, 0272.

63. The Examiner issued another Final Office Action on October 3, 2012, again rejecting all the pending claims based on the same rejections in the Non-Final Office Action dated February 14, 2012. The Examiner was not convinced by the Applicant's argument and asserted that the primary reference Waldorf explicitly taught the two levels of interactivity and Tolle and Tolle⁸¹⁹ taught the chart being interactive and chart template store. EX1002, 0292-93.

64. The Examiner suggested that the Applicant focus on features of chart tool box user interface and proposed the following amendment:

Examiner Proposed Amendment for Claim 7

(Proposed Amended) A computer-implemented method comprising:

receiving a selection, on a first website, of at least one interactive chart from a plurality of interactive charts; and displaying, in response to the selection the at least one interactive chart embedded in a chart box on a second website wherein the chart box is a flash widget,

wherein the at least one interactive chart is generated on a generation interface configured to allow a user to generate the interactive chart by with a chart template from a chart template store, and

wherein the at least one interactive chart, when displayed on the second website in the absence of the generation interface, enables a user to a) render the chart, b) obtain input from a user and c) render the chart using the input thereby enabling the interactive chart to retain interactivity

wherein the selection is made by dragging and dropping the at least one interactive chart from the plurality of interactive charts displayed in charting pane into a chart box folder connected to the chart box such that the retrieve code for data feed displays the interactive charts on the first website wherein the interactive chart is automatically displayed on the second website using a data feed from the first website to the second website upon said dragging and dropping selection.

EX1002, 0293.

65. On December 19, 2012, the Applicant amended independent claim 7 based on the Examiner's suggestion and canceled claims 1-6, 8, and 12-24. EX1002, 0306-09. On January 16, 2013, the Examiner issued an Advisory Action, rejecting claims 7 and 9-11. EX1002, 314-15.

66. The Applicant then filed a Request for Continued Examination (RCE) on January 22, 2023 to have the after-final amendments entered. EX1002, 0319-320.

67. The Office then issued a Notice of Allowance. EX1002, 0325-330. The '000 patent issued on August 27, 2013. EX1002, 0361.

C. Provisional Application No. 61/077,728

68. The '000 patent claims priority to U.S. Provisional Application No. 61/077,728 ("the '728 provisional application"), which was filed on July 2, 2008. EX1001, (60). Exhibit 1013 is a copy of the '728 provisional application.

69. To be entitled to the benefit of the filing date of the '728 provisional application, I understand that the '728 provisional application must provide sufficient written description for the claimed invention in the '000 patent. In other words, I understand that the '728 provisional application must describe the claimed invention in the '000 patent in sufficient detail that one skilled in the art can reasonably conclude that the inventor had possession of the claimed invention.

70. I have reviewed the '728 provisional application and have concluded that a POSA would not have reasonably concluded that the inventors (Seymour Duncker and Tyron Montgomery) had possession of any claim in the '000 patent. The '728 provisional application lacks a complete concept of the later claimed invention. As I explain in this section, iCharts added new concepts into the claims after the '728 provisional application's filing date.

71. The specification in the '728 provisional application was very short (13 paragraphs, a little over one page) and different than the '000 patent. *See* EX1013, 0045-46. The '728 provisional application includes some figures that resemble figures in the '000 patent. *See* EX1013, 0054-073. However, most of the figures are completely different. Some figures, for example, are missing from the '728 provisional application (e.g., Figures 1, 2, 8, 11-46 of the '000 patent are not in the '728 provisional application).

72. There are key omissions from the '728 provisional application that were later claimed in the '000 patent. For example, the '728 provisional application does not describe "*a chart box*," let alone "*a chart box folder*" connected to the chart box such that retrieve code for data feed displays the at least one interactive chart on the first website." Rather, the inventors added these concepts to the alleged invention *after* the '728 provisional application.

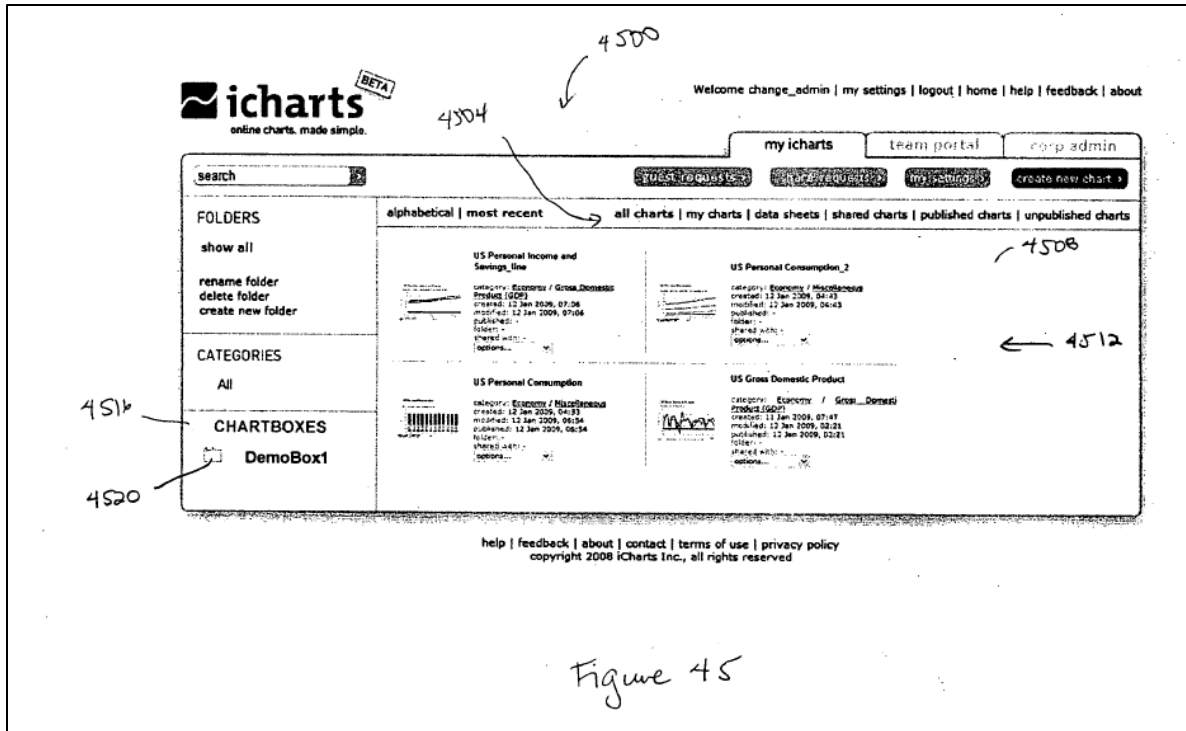
73. iCharts introduced the chart box and chart box folder in Figures 43-45 and paragraphs [0140] – [0145] of the filed specification on February 17, 2009. EX1002, 0043-45 (specification pages 34-36), 0109-111. Figure 45 and the portions of the specification that describe the chart box and chart box folder are reproduced below.

[0139] Figure 43 illustrates an exemplary screen shot of a business services application website 4300 according to one embodiment of the invention. The illustrated website 4300 includes a chart box 4304, shown in further detail in Figure 44. Referring to Figures 43 and 44, the chart box 4304 is configured to be embedded on a website to display one or more interactive charts on that website.

EX1002, 0042 (specification page 33).

[0142] A user may select possible interactive charts for inclusion in the chart box 4304 using the toolbar 4504. The user may use a charting pane 4508 to view and/or browse the interactive charts 4512 that can be selected for display in the chart box 4304. The user interface also includes a chart box region 4516 that includes a chart box folder 4520. The chart box folder 4520 is connected to the chart box 4304 such that the retrieve code for data feed displays the interactive chart(s) selected by the user in the chart box folder 4520.

EX1002, 0043-44 (specification pages 34-35).



EX1002, 0111.

74. This description of the chart box and chart box folder was not in the '728 provisional application, nor were they described in any of the figures of the '728 provisional application. See EX1013, 0045-46, 0053-073.

75. The claims of the '000 patent include additional elements that are not in the '728 provisional application. The '728 provisional application does not describe selecting at least one interactive chart “by *dragging and dropping the at least one interactive chart* from the plurality of charts into a chart box folder” as claimed in the '000 patent. See EX1013, 0045-46. The '728 provisional application merely states that a user can “drag and drop the data into a chart in a special

charting pane.” EX1013, 0045. Again, this concept was later added to the ’000 patent and its figures.

76. Nor does the ’728 provisional application describe that the at least one interactive chart *“is automatically displayed on the second website using a data feed from the first website to the second website upon the dragging and dropping selection”* as claimed in the ’000 patent. Rather, iCharts added the new feature when submitting the patent application 12/372,696 that issued as the ’000 patent. See EX1002, 0044-45. The portion of the specification that describes the feature of automatically displaying the interactive chart in the second website, when the interactive chart was selected in the first website by the dragging and dropping action is reproduced below:

[0142] The user can select the interactive chart(s) for display in the chart box 4304 by, for example, dragging and dropping one or more of the charts 4512 onto the chart box folder 4520. When the user drops the selected chart(s) 4512 onto the chart box folder 4520, the chart is automatically displayed in the chart box 4304 embedded on the user's website.

EX1002, 0044 (specification page 35).

77. iCharts also added new features for the interactive charts, including *“displaying a plurality of interactive charts embedded in a plurality of chart boxes on the second website.”*

78. As these examples show, the '728 provisional application does not provide sufficient written description of any of the '000 patent claims.

D. New Matter in U.S. Application No. 12/372,696

79. U.S. Application No. 12/372,696 (filed on February 17, 2009 and issued as the '000 patent) is a continuation-in-part (CIP) application of U.S. Application No. 12/205,802 (filed on September 5, 2008 and issued as U.S. Patent No. 8,271,892). EX1001, (63), 1:7-14. I understand that a CIP is an application containing a portion or all of the disclosure of an earlier application together with added matter not present in the earlier application.

80. I understand that there is simply no reason to presume that claims related to a CIP application are entitled to the effective filing date of an application filed before the CIP application. For example, to be entitled to the filing date of the '802 application (September 5, 2008), the '802 application must provide sufficient written description for the full scope of the claimed invention in the '000 patent. In other words, I understand that the '802 application must describe the claimed invention in the '000 patent in sufficient detail that one skilled in the art can reasonably conclude that the inventor had possession of the claimed invention.

81. I have reviewed the '802 application and determined that a POSA would not have reasonably concluded that the inventors had possession of the full

scope of any claim in the '000 patent as of September 5, 2008. In other words, the '802 application lacks a complete concept of the later-claimed invention.

82. iCharts added new concepts after the '802 application's filing date. For example, the '696 application includes three figures (i.e., Figures 43-45), which are not in the '802 application. EX1002, 0109-111.

83. This new matter led to the claimed feature of a computer-implemented method that *“receive[s] a selection, on a first website, of at least one interactive chart from a plurality of interactive charts; . . . wherein the selection is made by dragging and dropping the at least one interactive chart from the plurality of charts into a chart box folder connected to the chart box such that retrieve code for data feed displays the at least one interactive chart on the first website, and wherein the at least one interactive chart is automatically displayed on the second website using a data feed from the first website to the second website upon the dragging and dropping selection.”* EX1001, 18:23-48.

84. To the extent the specification describes these features in sufficient detail, the support lies within the new matter in paragraphs [0139]-[0145] and Figures 43-45 of the '696 application, which were not in the '802 application. EX1002, 0042-45 (specification pages 33-36), 0109-111.

E. Level of Ordinary Skill in the Art

85. A person of ordinary skill in the art (“POSA”) in the technical field of the ’000 patent at the relevant time would typically have had: a B.S. degree in Computer Science, Computer Engineering, or equivalent experience relating to computing, as well as two to three additional years of academic or industry experience related to web application development and data visualization services. The POSA would have, therefore, been familiar with client-side scripting technologies, such as JavaScript and/or ActionScript. The POSA would have further worked as part of a multidisciplinary team and drawn upon not only her own skills, but of others on the team, e.g., to solve a given problem.

86. At the time of the alleged invention (2009), I was familiar with this level of skill. I would have qualified (and presently qualify) as at least a POSA.

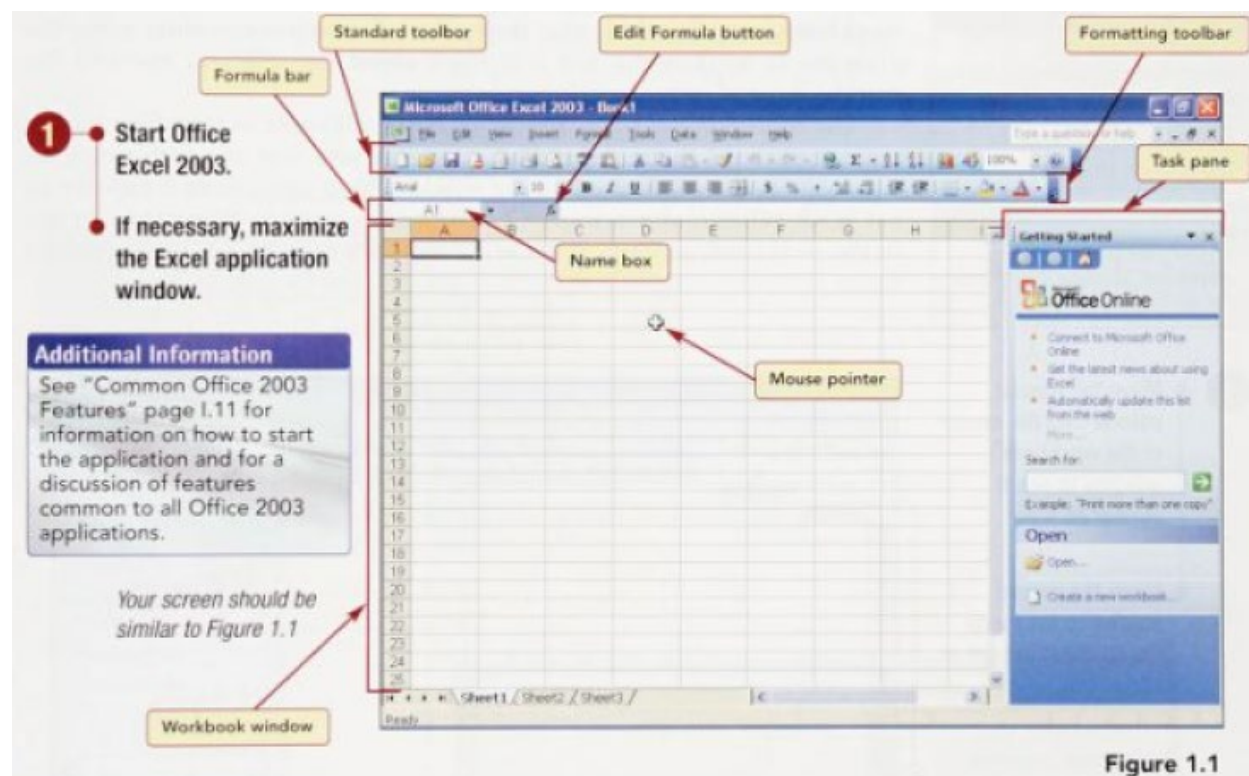
VI. Technology Overview

87. As I explain in the following sections, data visualization tools existed long before the ’000 patent, as did interactive charts, data/chart panes, and drag-and-drop techniques.

A. Data Visualization Tools

88. The ’000 patent itself recognizes that data visualization tools such as Microsoft Excel existed before the alleged invention. *See* EX1001, 1:23-24 (“Microsoft Excel is typically used to enter data into spreadsheets and sometimes make charts or graphs from that data.”). At the time, Microsoft Excel 2003 allowed

users to input data, insert formulas to alter and create new data, and generate interactive charts. The user interface for an Excel 2003 workbook is shown below.



EX1011, 0044 (Figure 1.1).

89. The area including the rows and columns is commonly referred to as a spreadsheet. The spreadsheet is where the user enters data. EX1011, 0046. Each space within the spreadsheet is a cell, identified by the column of the cell followed by the row of the cell. EX1011, 0047. Cells support textual data, numeric data, and formulas that perform calculations on data within cells. EX1011, 0053.

90. One of Excel's strengths is its ability to generate visualizations such as charts and graphs from the data entered into the spreadsheet. Excel 2003 offered fourteen types of charts for users to choose from. EX1011, 0079.

Concept 3

Charts

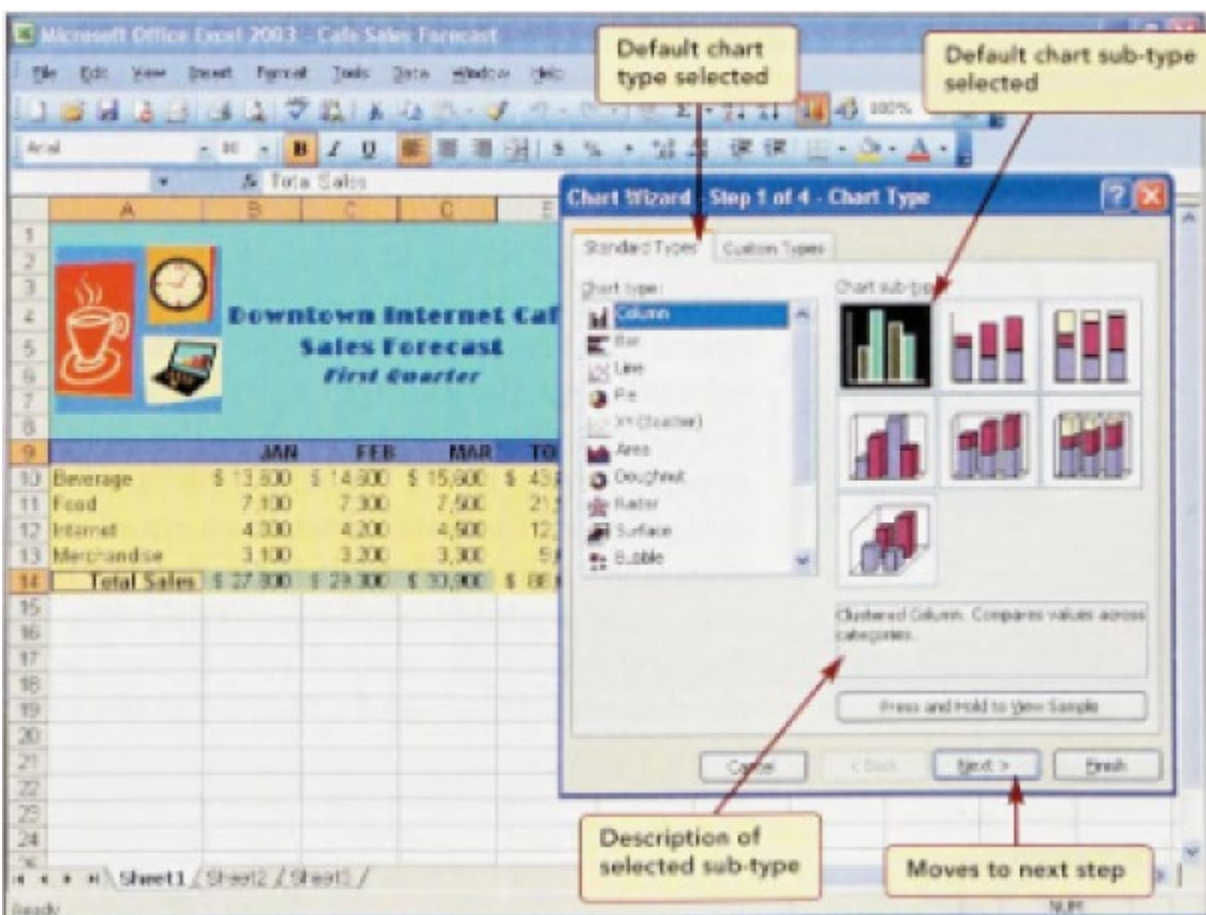
3 A **chart** is a visual representation of data that is used to convey information in an easy-to-understand and attractive manner. Different types of charts are used to represent data in different ways. The type of chart you create depends on the type of data you are charting and the emphasis you want the chart to impart.

Excel 2003 can produce 14 standard types of graphs or charts, with many different sub-types for each standard type. In addition, Excel includes professionally designed, built-in custom charts that include additional formatting and chart refinements. The basic chart types and how they represent data are described in the following table.

Type	Description	Type	Description
	Area charts show the magnitude of change over time by emphasizing the area under the curve created by each data series.		Radar charts display a line or area chart wrapped around a central point. Each axis represents a set of data points.
	Bar charts display data as evenly spaced bars. The categories are displayed along the Y axis and the values are displayed horizontally, placing more emphasis on comparisons and less on time.		XY (scatter) charts are used to show the relationship between two ranges of numeric data.
	Column charts display data as evenly spaced bars. They are similar to bar charts, except that categories are organized horizontally and values vertically to emphasize variation over time.		Surface charts display values in a form similar to a rubber sheet stretched over a 3-D column chart. These are useful for finding the best combination between sets of data.
	Line charts display data along a line. They are used to show changes in data over time, emphasizing time and rate of change rather than the amount of change.		Bubble charts compare sets of three values. They are similar to a scatter chart with the third value displayed as the size of bubble markers.
	Pie charts display data as slices of a circle or pie. They show the relationship of each value in a data series to the series as a whole. Each slice of the pie represents a single value in the series.		Stock charts are high-low-close charts. They require three series of values in this order.
	Doughnut charts are similar to pie charts except that they can show more than one data series.		Cylinder charts display values with a cylindrical shape.
			Cone charts display values with a conical shape.
			Pyramid charts display values with a pyramid shape.

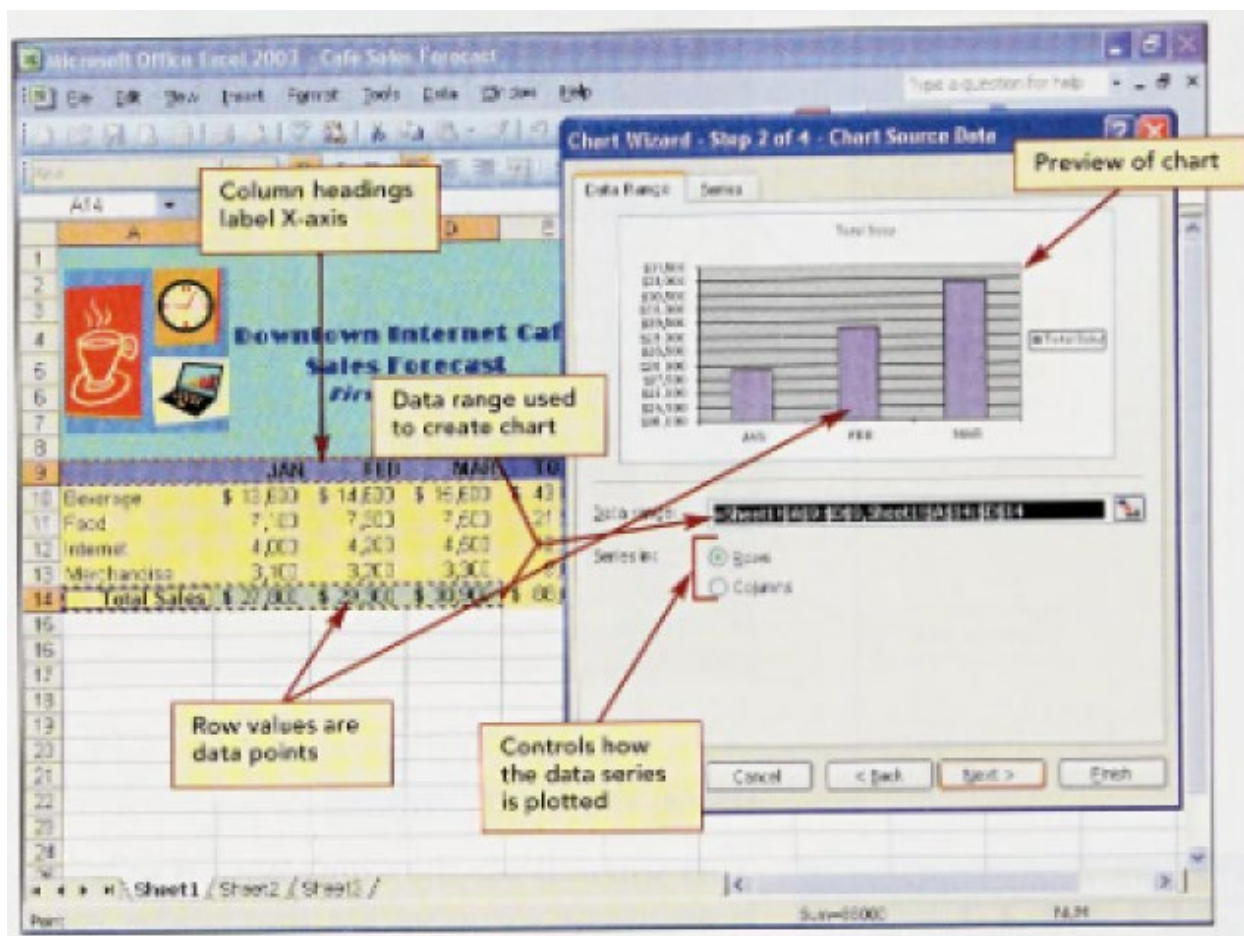
EX1011, 0079.

91. Excel 2003 allowed users to select data they wished to chart, and then use a Chart Wizard to generate the chart, as shown below. EX1011, 0082.



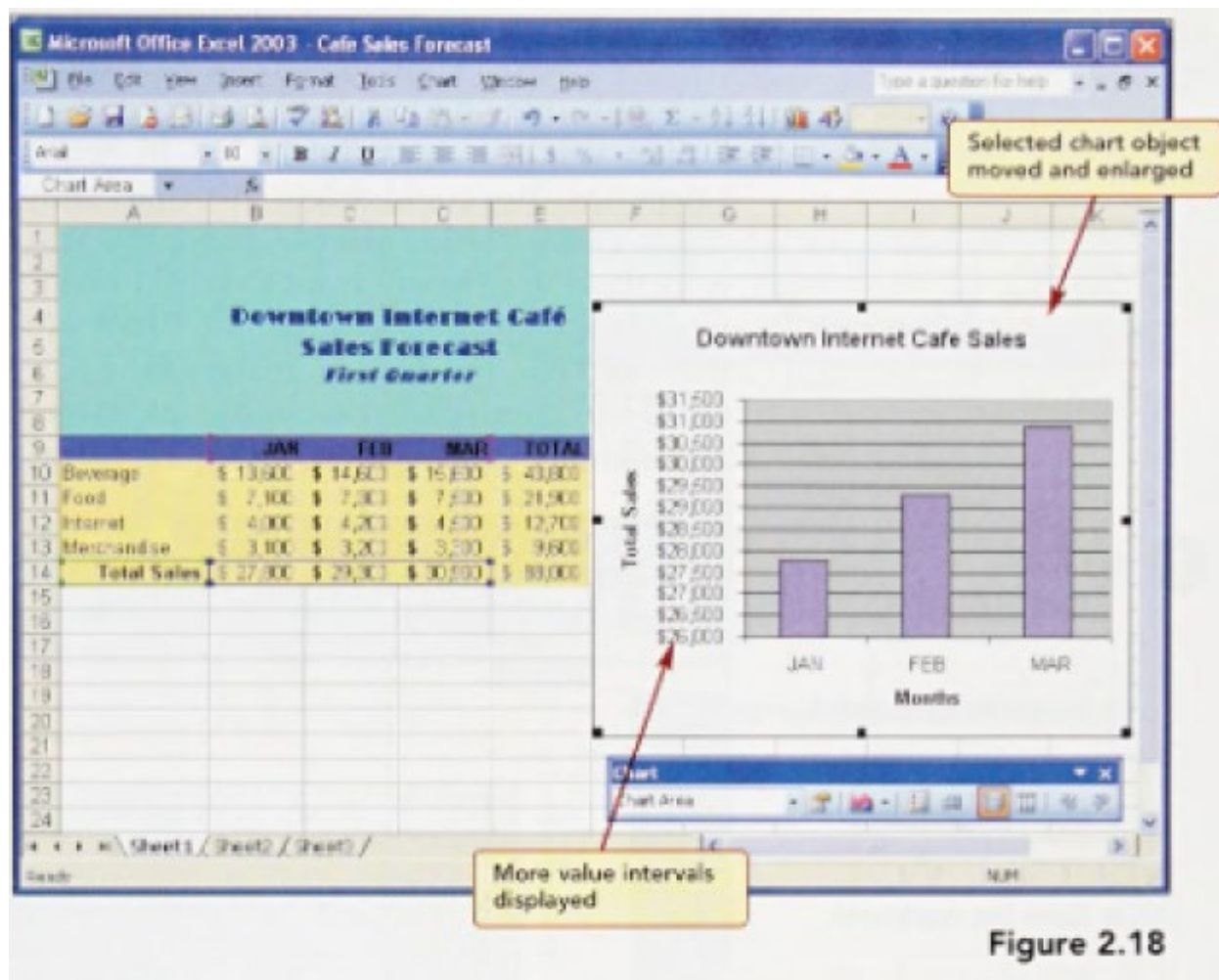
EX1011, 82 (Figure 2.11).

92. The Chart Wizard allowed users to select a chart type and the cells used to populate the chart. EX1011, 0083. The Chart Wizard also allowed users to customize part of the chart such as the title, axes, and legend. EX1011, 0084.



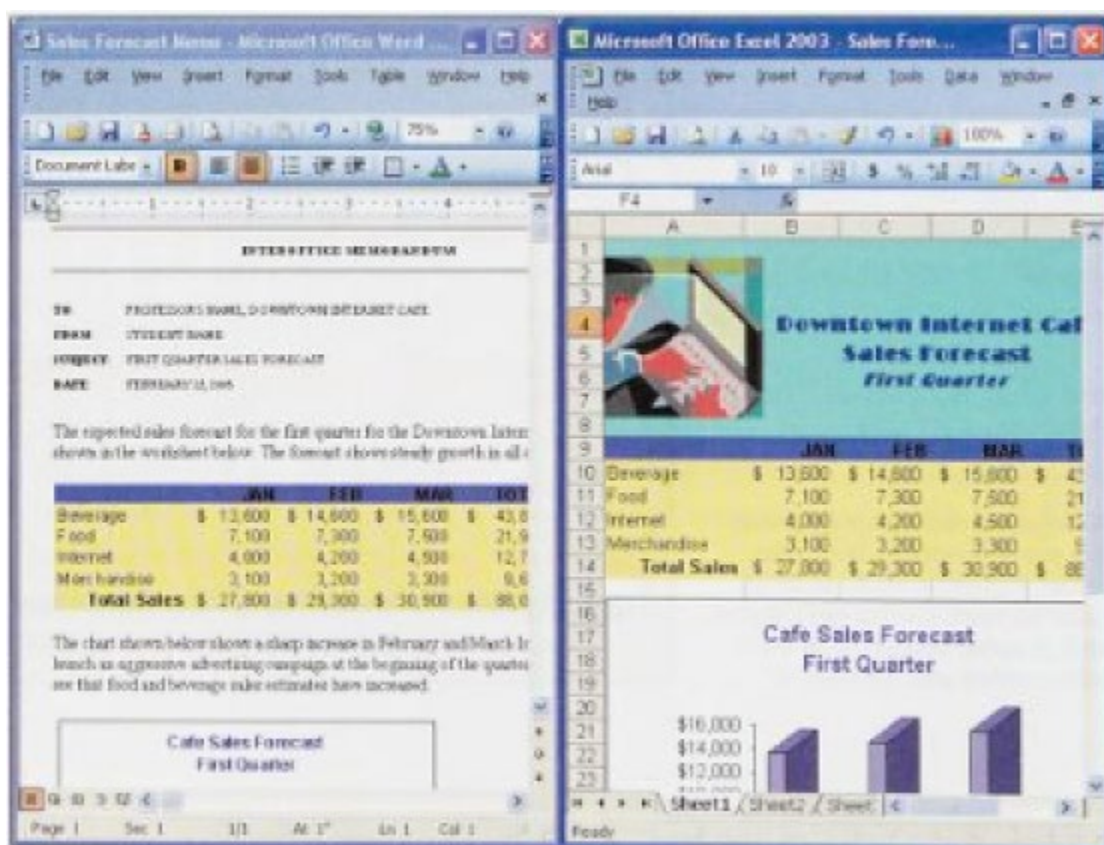
EX1011, 0083 (Figure 2.12).

93. Once created, Excel 2003 allowed users to interact with the chart. Perhaps the simplest interaction is that the user could use the mouse to drag (e.g., move) the chart to a new location. EX1011, 0087. The user could also use the mouse to change the size of the chart. EX1011, 0087. The chart would react to the size change by updating displayed data. For example, additional values may be displayed along an axis when the chart size was increased. EX1011, 0088.



EX1011, 0088 (Figure 2.18).

94. Another benefit of Excel 2003 was that it easily integrated with other Microsoft Office products such as Microsoft Word and Microsoft PowerPoint. For example, as shown below, charts and graphs generated in Excel could be copied into Microsoft Word documents.



EX1011, 0019.

95. Tools such as PowerPoint also included their own visualization tools. PowerPoint 2003 allowed users to link a PowerPoint chart to an Excel spreadsheet. EX1012, 0022. As a result, when the user updates the spreadsheet in Excel, the linked chart in PowerPoint automatically updates. EX1012, 0023.

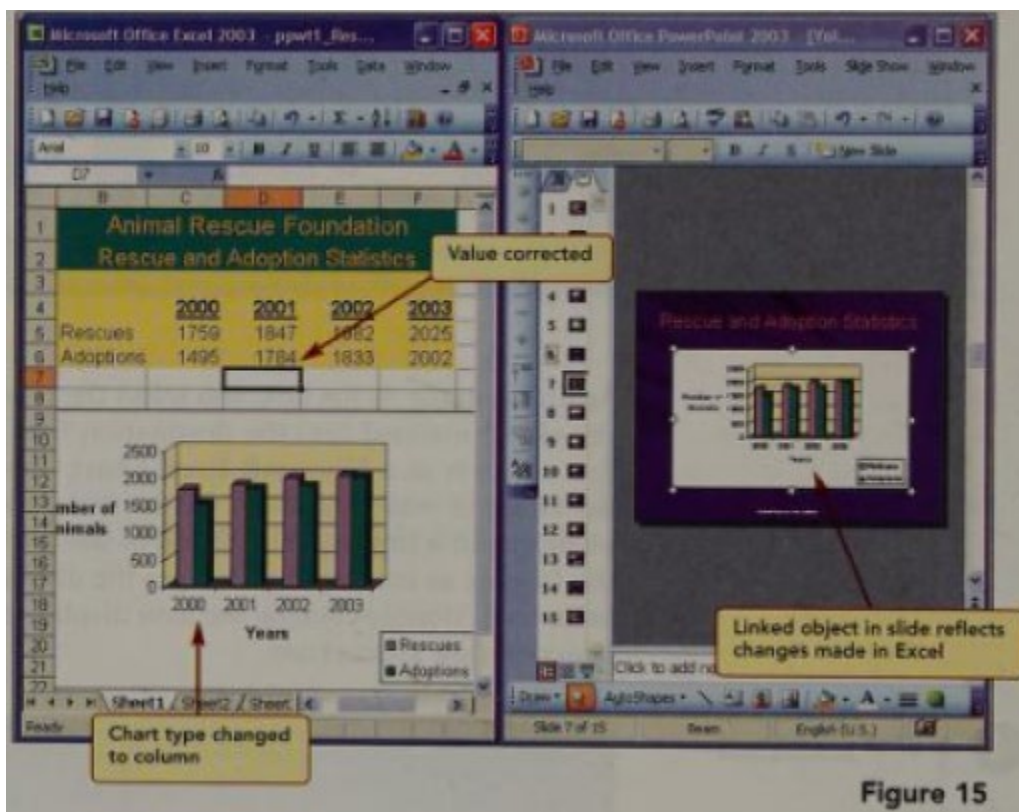
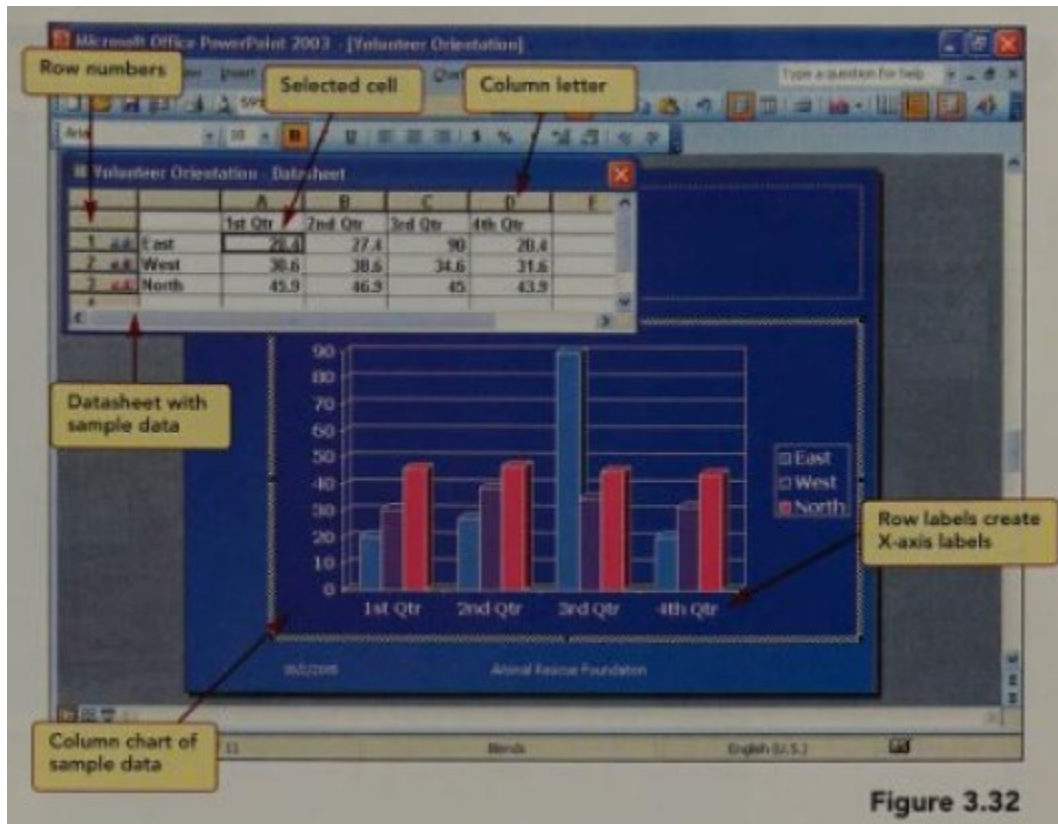


Figure 15

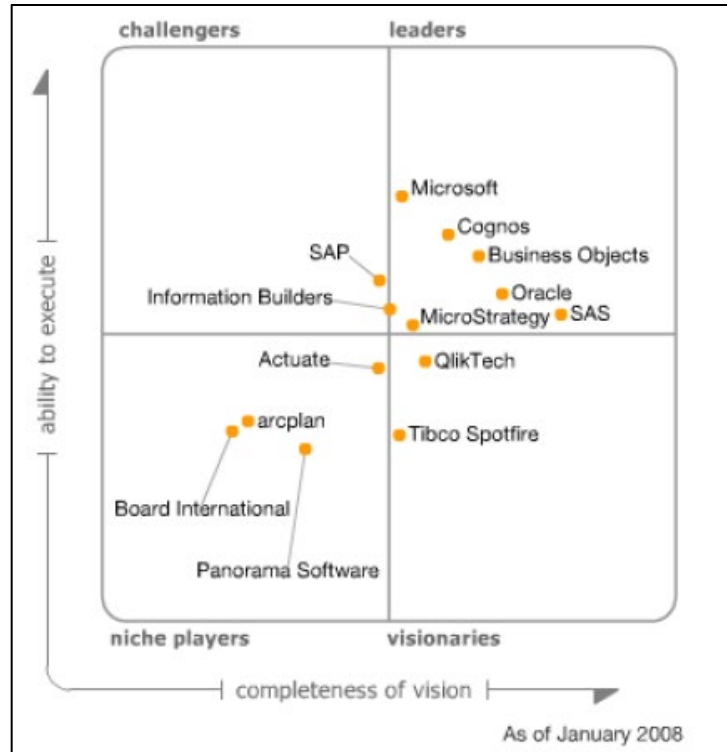
EX1012, 0025 (Figure 15).

96. PowerPoint 2003 also offered its own charting and graphing tool. The built-in graph tool allowed PowerPoint users to generate and customize charts without having to access an additional program. PowerPoint's built-in graph tool included many of the same features as Excel, such as chart type selection, data selection, customizable chart axes, customizable chart shapes, and customizable chart colors. EX1012, 0063-092. An example of the built-in chart tool is below.



EX1012, 68 (Figure 3.32).

97. The power of visualization tools such as Excel 2003 and PowerPoint 2003 did not go unnoticed. As businesses recognized the utility of data visualization tools, companies began creating competing visualization software. As of January 2008, many companies had brought to market improved business intelligence software for data visualization, some of which are noted in the below figure.

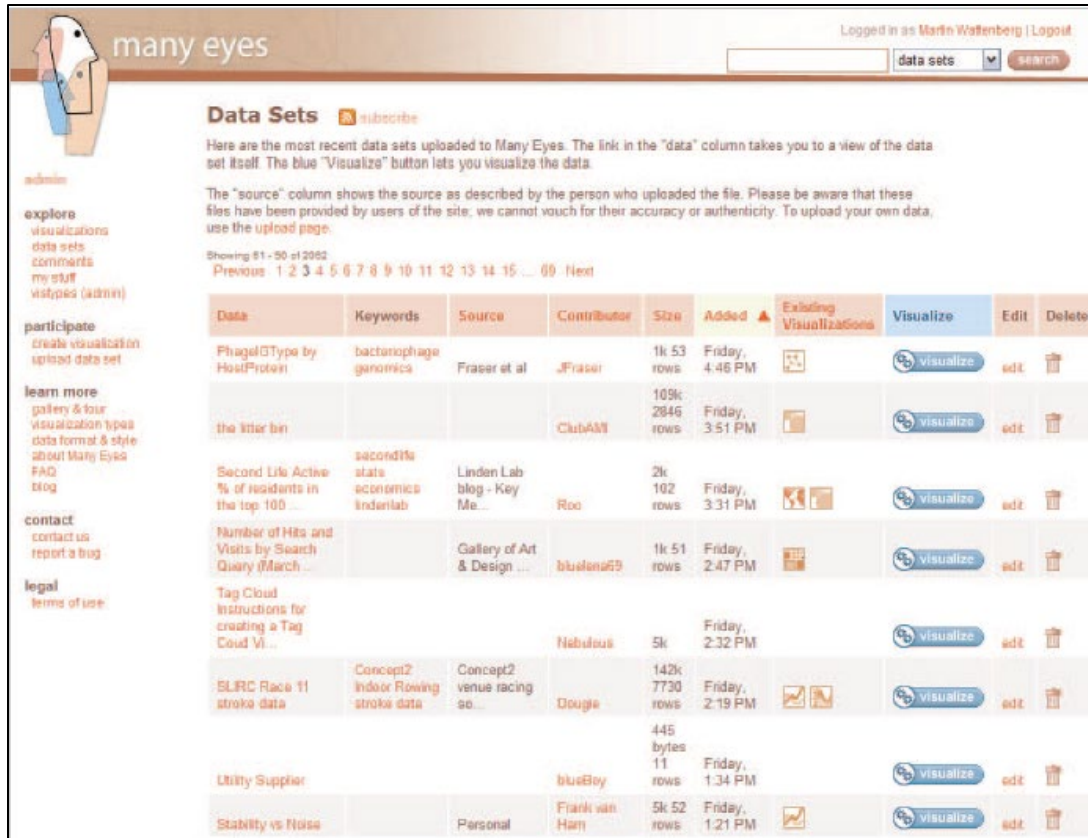


EX1028, 0002 (Figure 1).

98. Companies such as Microsoft, Cognos, Oracle, SAP, SAS, Tibco, and QlikTech started competing for market share by selling business intelligence platforms with different capabilities. One of these capabilities was advanced visualization, which provided users “the ability to display numerous aspects of the data more efficiently by using interactive pictures and charts, instead of rows and columns.” EX1028, 0005. As recognized in the art, “[n]ot only are interactive visualizations a key medium for communicating in a data-rich world, but preliminary reports hint that visualizations potentially have a catalytic effect on storytelling [] and collective data analysis [].” EX1039, 0001.

99. Data visualization tools on the market included sophisticated dashboards, web-based publishing, Microsoft Excel integration, and interactive graphics and charts. EX1028, 0005. These tools had widespread use across many different industries. For example, Cognos was purchased by IBM in 2007 and was used by companies such as Home Depot, Amazon, and 7-Eleven. EX1038, 0001. In 2008, SAP's NetWeaver business intelligence product had over 13,000 deployments. EX1028, 0018. QlikTech's product, QlikView, had over 7,000 customers in more than 80 countries. EX1028, 0017.

100. IBM was also developing a website called "ManyEyes" to allow Internet users to create and share data and charts with other Internet users. *See generally* EX1039. With ManyEyes, users could upload and share data. EX1039, 0003. Shown below, users could see a list of datasets added to the website:



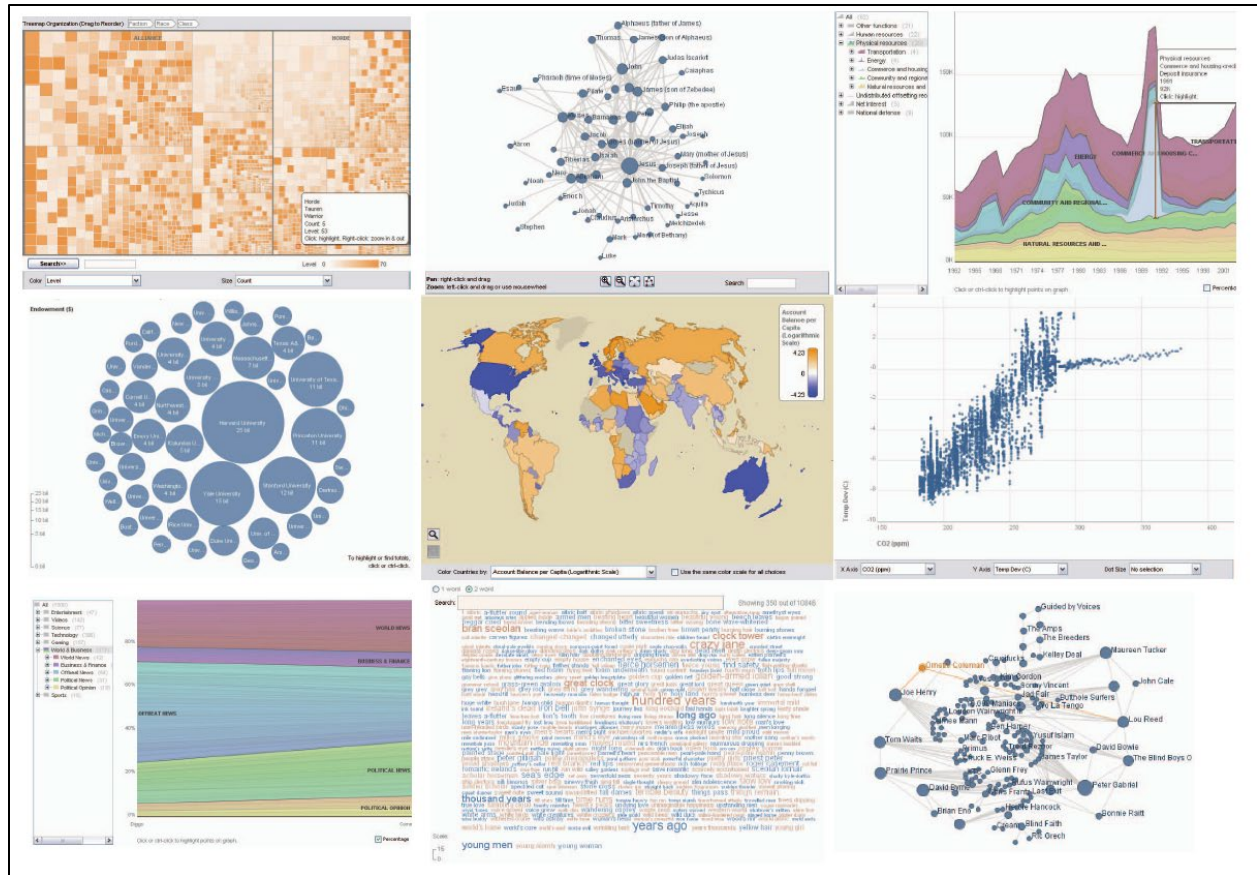
The screenshot shows the ManyEyes website interface. At the top, there is a header with the ManyEyes logo, a search bar, and a login status for 'Martin Wallenberg'. Below the header, there is a sidebar with navigation links: 'admin', 'explore' (visualizations, data sets, comments, my stuff, visit pages (admin)), 'participate' (create visualization, upload data set), 'learn more' (gallery & tour, visualization types, data format & style, about ManyEyes, FAQ, blog), 'contact' (contact us, report a bug), and 'legal' (terms of use).

The main content area is titled 'Data Sets' with a 'subscribe' button. It contains a paragraph explaining that the list shows recent data sets uploaded to ManyEyes, and that the 'source' column shows the source as described by the person who uploaded the file. Below this, there is a table of data sets.

Data	Keywords	Source	Contributor	Size	Added	Existing Visualizations	Visualize	Edit	Delete
PhageGType by HostProtein	bacteriophage genomics	Fraser et al	JFraser	1k 53 rows	Friday, 4:46 PM		visualize	edit	delete
the litter bin			ClaraAM	109k 2646 rows	Friday, 3:51 PM		visualize	edit	delete
Second Life Active % of residents in the top 100 ...	secondlife stats economics lindenslab	Linden Lab blog - Key Me...	Roo	2k 102 rows	Friday, 3:31 PM		visualize	edit	delete
Number of Hits and Visits by Search Query (March ...		Gallery of Art & Design ...	blueleaves9	1k 51 rows	Friday, 2:47 PM		visualize	edit	delete
Tag Cloud Instructions for creating a Tag Cloud Vi...			Nebulous	5k	Friday, 2:32 PM		visualize	edit	delete
SLRC Race 11 stroke data	Concept2 indoor Rowing stroke data	Concept2 venue racing so...	Dougie	142k 7730 rows	Friday, 2:19 PM		visualize	edit	delete
Utility Supplier			blueBoy	445 bytes 11 rows	Friday, 1:34 PM		visualize	edit	delete
Stability vs Noise		Personal	Frank van Ham	5k 52 rows	Friday, 1:21 PM		visualize	edit	delete

EX1039, 0003 (Figure 1).

101. ManyEyes—like many data visualization tools at the time—provided different types of basic visualization components. EX1039, 0003. The following figure shows “[a] tiny cross-section of the visualizations users created by matching their own data with [ManyEyes’s] visualization components.” EX1039, 0004.



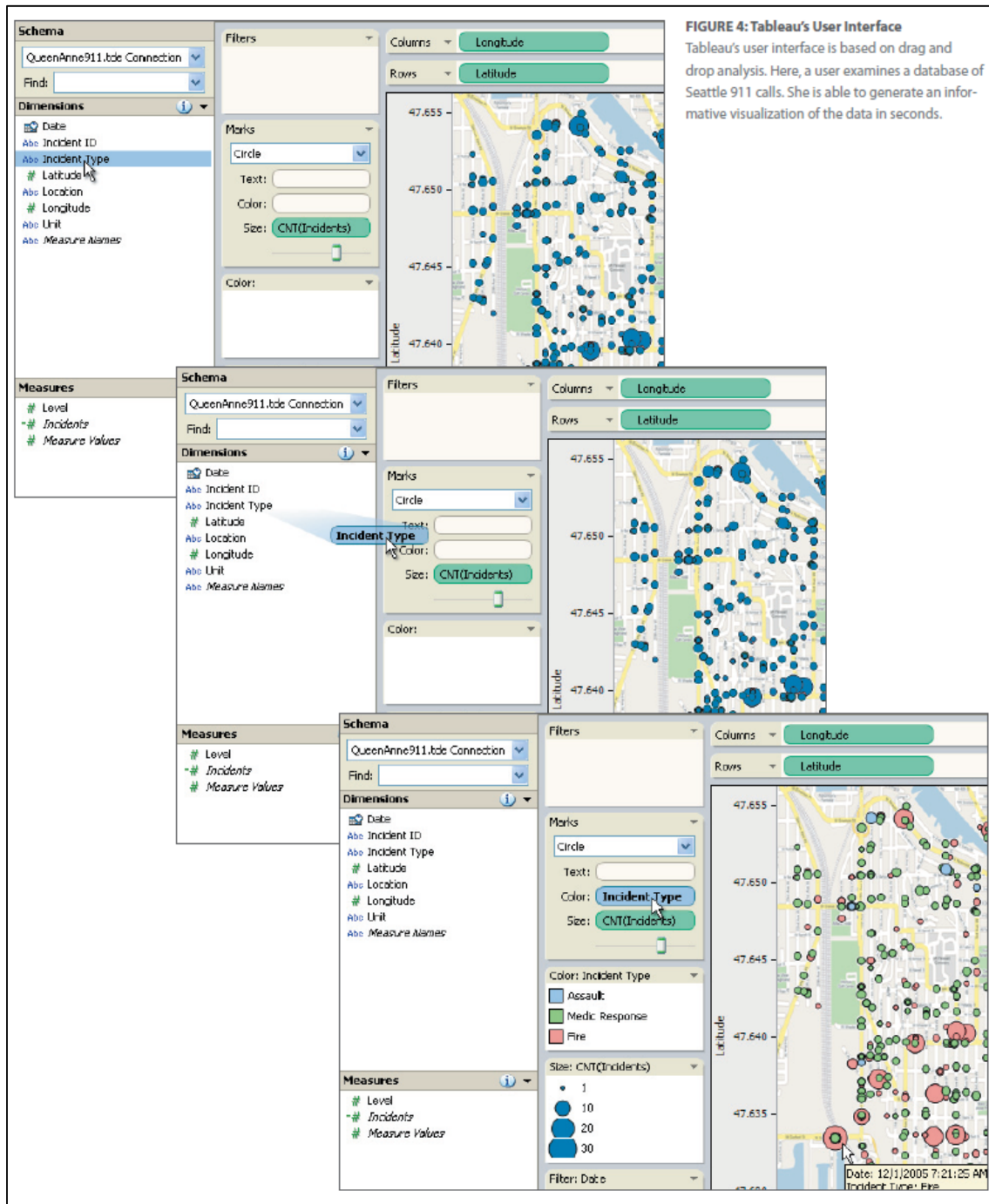
EX1039, 0004 (Figure 2).

102. The ManyEyes charts were interactive. EX1039, 0001, 0007. “This interactivity allow[ed] users to drill down into details, view the data from different perspectives and generally ma[de] the visualizations fun to use.” EX1039, 0007.

103. Each of the disclosures cited above included the ability to generate interactive charts with different interactive features. Interactive zoom features were known. See EX1011, 0115 (disclosing zooming in on an Excel 2003 worksheet); EX1012, 0050-51 (discloses zoom on PowerPoint 2003 slides); EX1039, 0004 (disclosing a zoomable interface); EX1035, 0003 (disclosing an interactive zoom

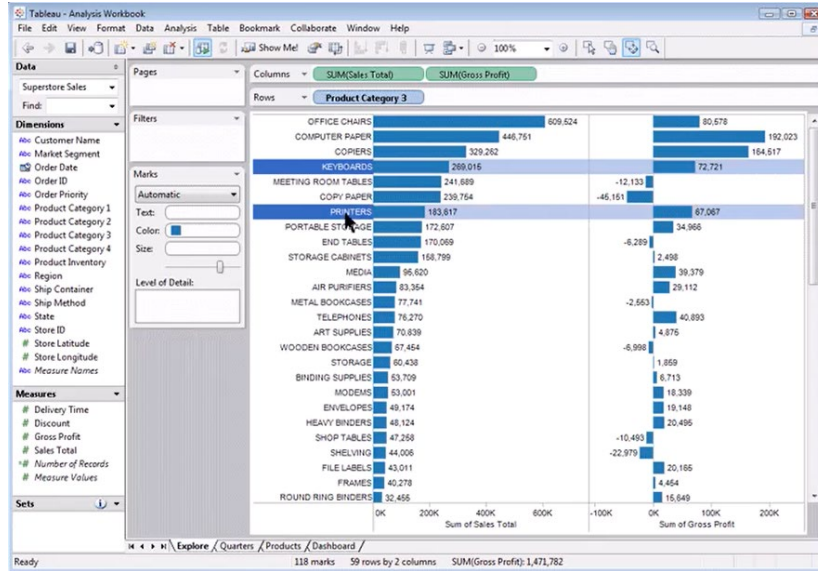
interface). Mouse-over pop-up windows were known. *See* EX1035, 0010 (discloses showing labels when the mouse is moved over corresponding data); EX1053 (incorporating mouse over pop-ups in an interactive graphic). Drill down features were known. *See* EX1039, 0007 (discloses interactive features allowing the user to drill-down into data details); EX1040, 0008-0010 (disclosing a drill-down feature through categorization of data graphics). Drag and drop features were known. *See* EX1040, 0002-03, 0009 (disclosing drag and drop to update a display); EX1035, 0003 (describing drag and drop to map attributes to shapes and labels).

104. Indeed, Tableau also had a system that generated charts:



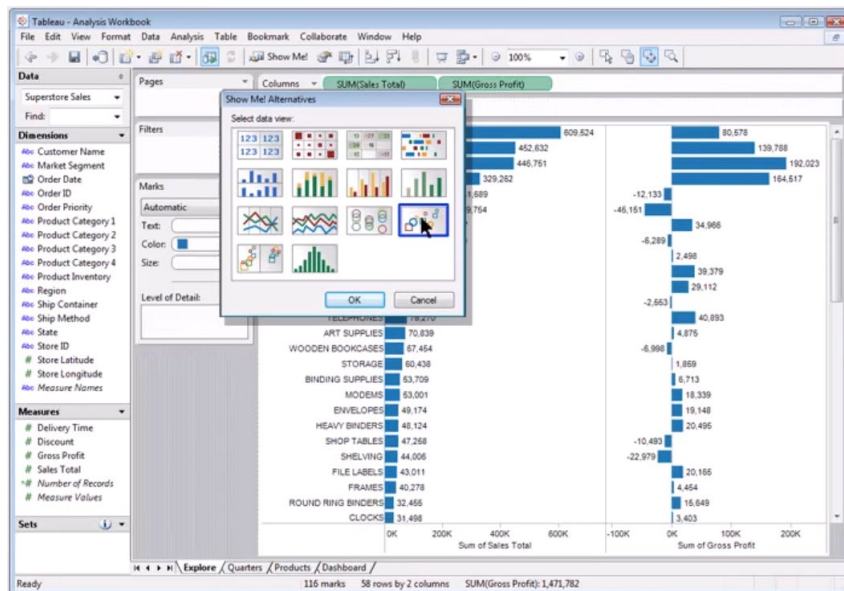
EX1041, 0012; *see also* EX1049.

105. With Tableau, users could select data and “change what you’re looking at as well as how you’re looking at it.” *See* EX1049, 00:12-51.



EX1049, 00:39.

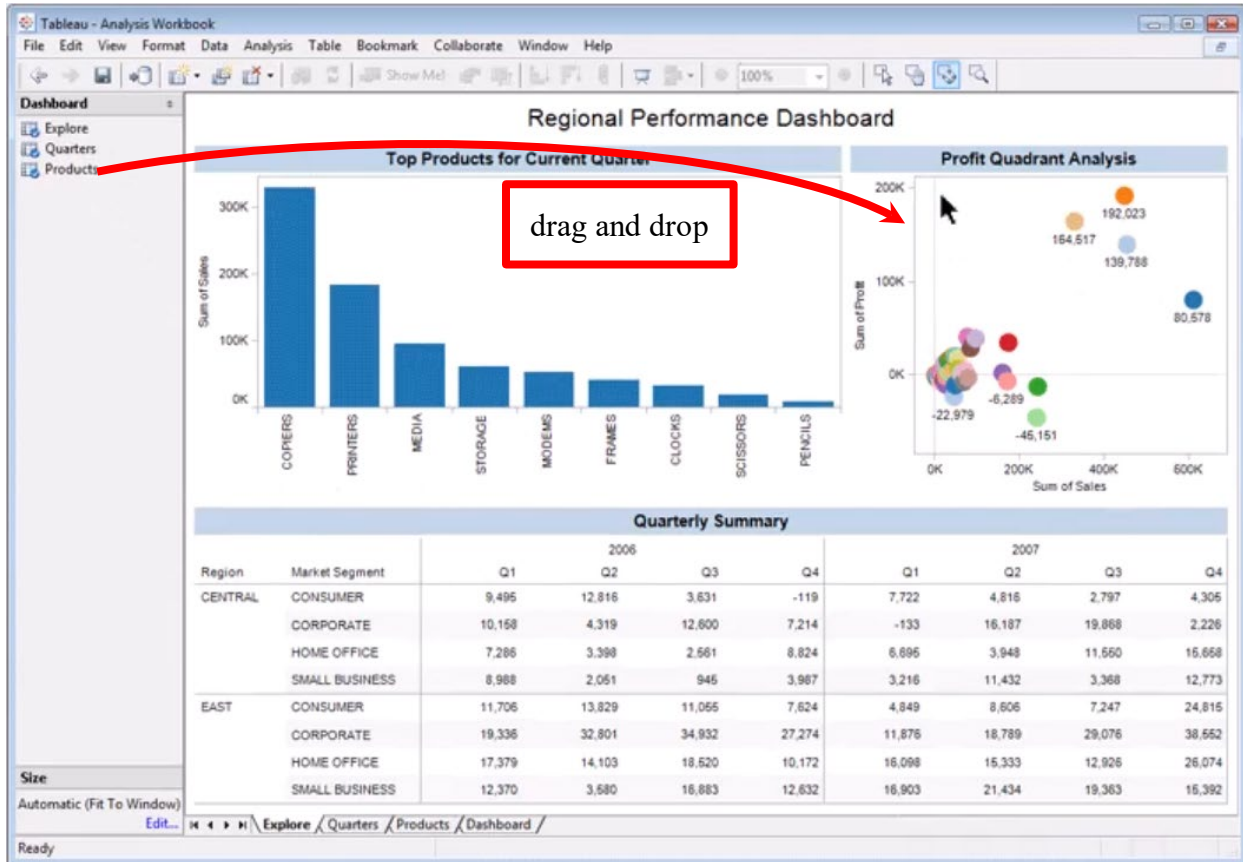
106. Shown below, after selecting which data the user wants to see, the user can select different chart types to visualize the data differently.



EX1049, 00:50.

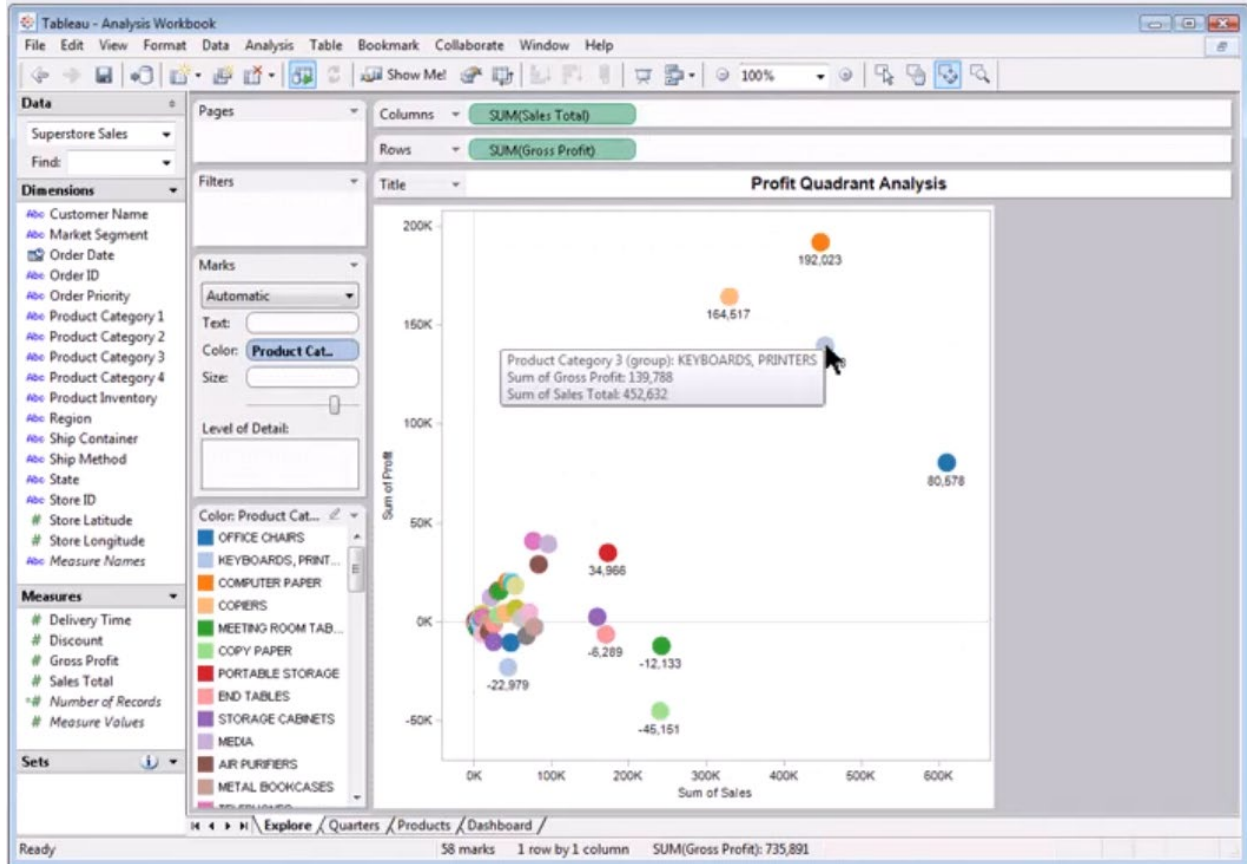
107. Tableau users could also add charts to dashboards for ongoing use.

EX1049, 00:59-1:04. “It’s as easy as drag and drop.” EX1049, 00:59-1:04. After the chart is dropped into a dashboard, a user could then publish the dashboard on the web, making it viewable to other users. EX1049, 1:04-1:12, 5:18-6:22.



EX1049, 1:04 (annotated).

108. After Tableau charts were published to a dashboard, the charts maintained interactive features. *See, e.g.,* EX1049, 4:38-6:21. As a video showing Tableau states, “these dashboards are live and interactive.” *See* EX1049, 4:41-4:43. One interactive feature in Tableau included mouse-over popups, as shown below:



EX1049, 00:53.

109. Accordingly, as known in the art, data visualization tools used to generate interactive charts were well known and conventional by 2008.

B. Flash

110. Flash was a widely known technology underlying data visualization technologies. Flash was created to overcome the performance issues associated with relying solely on HTML to create interactive websites.

111. Flash involves two components: a Flash file and a Flash player, which provides a runtime that executes and displays content from a Flash file. The first component, a Flash file, is created to deliver “vector graphics and animation over

the Internet.” EX1043, 0007. Flash files include vector graphics and code, such as ActionScript, interpreted by the client’s browser. EX1043, 0007, 0083; *see also* EX1052, ¶49 (disclosing ShockWave files including vector graphics). Flash files are constructed using a series of tags. Tags can be definitional, identifying data such as text, shapes, sounds, or bitmaps. EX1043, 0010. Each definitional tag has a unique identifier. EX1043, 0010. The unique identifier is assigned to the tag’s content within a dictionary. EX1043, 0010. This allows chunks of content associated with a given tag to be located and ordered. EX1043, 0010. Tags can also be control tags that manipulate content within definitional tags. EX1043, 0010. Control tag types include ShowFrame, PlaceObject, and RemoveObject. EX1043, 0027. Flash files are typically compressed prior to transmission. EX1043, 0012.

112. The second component is the Flash player. The Flash player resides on a client device, and decompresses and renders data from a Flash file. For example, when a ShowFrame tag is identified, the received content in the definitional tags is rendered on the display. EX1043, 0011; *see also* EX1051, ¶¶54-55 (discloses transmitting flash files for display between devices); EX1052, ¶¶25, 68, 72 (describes downloading and displaying flash movies on an electronic reading device). The Flash player keeps track of changes within received data. As a result, if content remains unchanged, the Flash player won’t change the rendered content. EX1043, 0012. The Flash Player can also use SharedObjects (i.e., cookies)

to store data for reuse. EX1042, 0008. SharedObjects allow the Flash Player to render content without a connection to a server, and also reduce the amount of communication required between the client and server. EX1042, 0008.

113. Using Flash, a developer can create highly customized, interactive interfaces. For example, using Flash a developer can create custom fonts, scroll bars, combo boxes, radio buttons, check boxes, and graphs. EX1042, 0009; *see also* EX1046, 5:19-33, 8:6-11, 10:42-44 (describing interactive flash objects); EX1052, ¶¶6, 7 (discloses receiving input at displayed flash files).

114. Once created and customized, these components can be animated using ActionScript. EX1042, 0009. Flash also allows developers to include MP3 audio and video within created components. EX1042, 0009-010.

C. JavaScript

115. JavaScript was also a widely known technology underlying data visualization technologies. JavaScript is an interpreted programming language that executes code on a client's browser. EX1044, 0015. Notably, this allows web pages to execute code on the client, supporting features such as user interaction, browser control, and dynamically created HTML. EX1044, 0015. Webpages are built using HTML defining the structure of the page. Webpage designers may include JavaScript within HTML, designed to execute within the client's browser when they access the page. EX1044, 0020. For example, instead of listing a range

of numbers to display within a webpage, a JavaScript function could be used to iterate and display each number within the range. EX1044, 0020.

116. One benefit of JavaScript is its ability to use event handlers responsive to user inputs. For example, a web developer can create two buttons, each listening for a mouse click, and in response, call separate functions when the click is detected. EX1044, 0035. Thus, using JavaScript, a web developer can define functionality based on events such as single mouse clicks, double mouse clicks, mouse movements, keys pressed, keys pressed down, and keys released. EX1044, 0037. Event handlers can also be configured for the user's interactions with displayed elements. For example, event handlers can listen for a window resize, selection of text, form submissions, and element selection. EX1044, 0037.

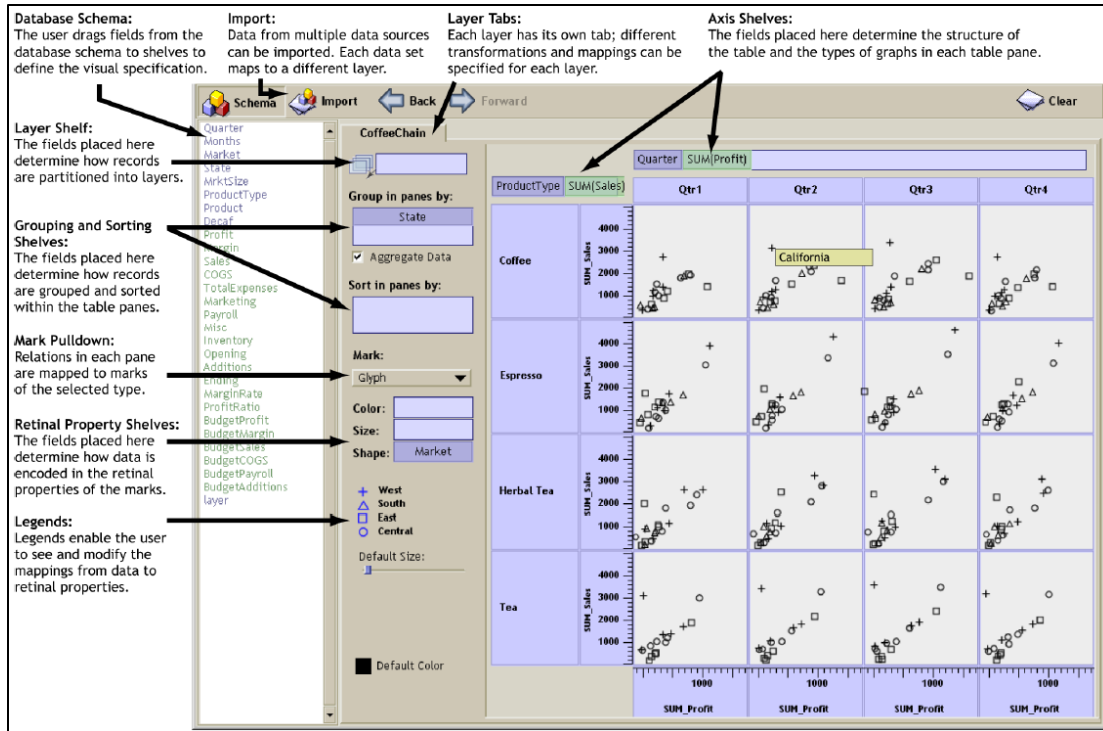
117. JavaScript can also be used within a framework known as asynchronous JavaScript and XML (AJAX). EX1057, 0025. In AJAX, the browser submits a request for content from the server using an XML HTTP request. EX1057, 0040, 0043. The developer then defines a function to call when the response is received. EX1057, 0047. In AJAX, the request occurs in the background, allowing the user to continue interacting with the web page while the server responds. EX1057, 0040. Thus, although the web page is being updated with content from the server, the client doesn't have to reload an entirely new instance of the webpage each time it receives data from the server.

D. Drag and Drop

118. Jef Raskin, the “renaissance man” behind Apple’s 1984 Macintosh, is credited with creating many features of the graphical user interface we know and use today. EX1037, 0001. “Besides overseeing the development [of the Macintosh], Raskin is considered responsible for the machine’s drag-and-drop feature, a user-friendly innovation later adopted by other computer operating systems, including Microsoft Windows.” EX1037, 0001. Leveraging a one-button mouse, Raskin allowed contiguous areas of the screen to be selected by depressing the mouse button, dragging the mouse, and releasing the button. EX1036, 0013. This technique, known as click and drag on the Macintosh, has been replicated countless times since its implementation on the Macintosh. EX1036, 0013.

119. Fast forward from 1984 to 2008, drag and drop had become ubiquitous in most, if not all graphical user interfaces. As I explained above, both Excel 2003 and PowerPoint 2003 used drag and drop to move and resize displayed charts. *See* EX1011, 0087-88 (moving a chart within Excel 2003 via drag and drop).

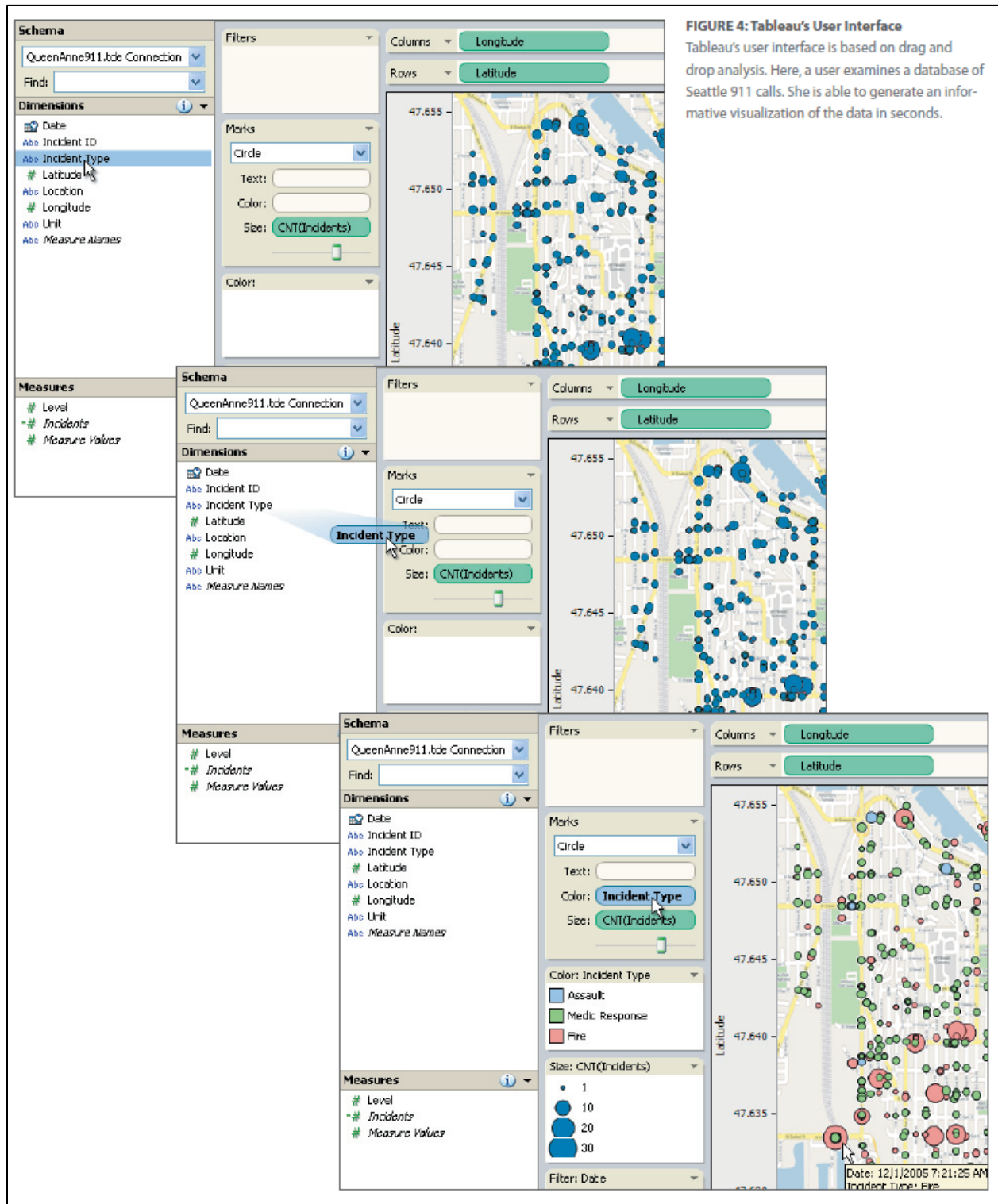
120. Another system called Polaris used drag and drop to “construct table-based displays of relational data.” EX1040, 0003. Shown below, “[t]he user drags fields from the database schema to shelves to define the visual specification.” EX1040, 0003 (Fig. 1).



EX1040, 0003 (Fig. 1).

121. ManyEyes also included drag and drop: “We designed a widget that allows user to reorder column names by drag and drop, again, changing the visualization on the fly.” EX1039, 0005.

122. Tableau also used drag-and-drop:



EX1041, 0012; *see also* EX1049.

123. Numerous prior art patents also used drag and drop to move and alter data. *See* EX1045, Abstract (discloses editing database data using drag and drop techniques); EX1046, 8:6-11 (describing drag and drop operations in Flash applications); EX1047, 6:39-44 (discloses dragging an icon of a source view onto a target view to display data in the target view when the source view icon is selected); EX1048, 8:59-66, FIGs. 3A, 3B (adding a field to a view by drag and drop); EX1050, Abstract, FIG. 2 (drag and drop within a web browser); EX1054, 2:20-27 (using drag and drop to transfer data between charts); EX1055, 1:61-67, FIG. 3a (charting data using drag and drop). As these patents show, drag and drop had allowed users to transfer data within a program for many years before 2008.

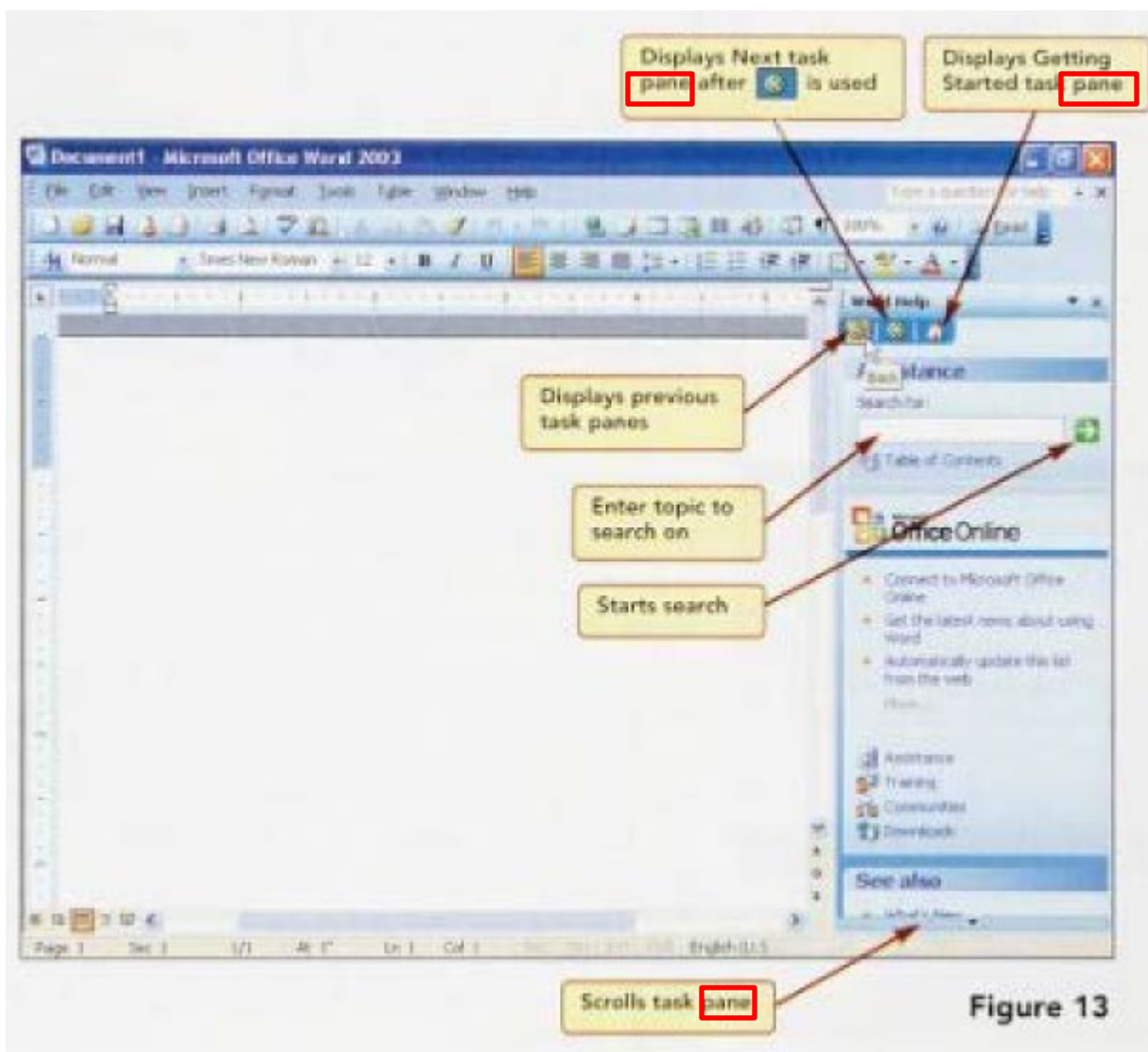
124. By 2008, drag-and-drop had become a basic, well-known design choice in graphical user interfaces.

E. Panes

125. Another basic, well-known design choice in graphical user interfaces was the use of different panes, e.g., one pane for data and one pane for charts.

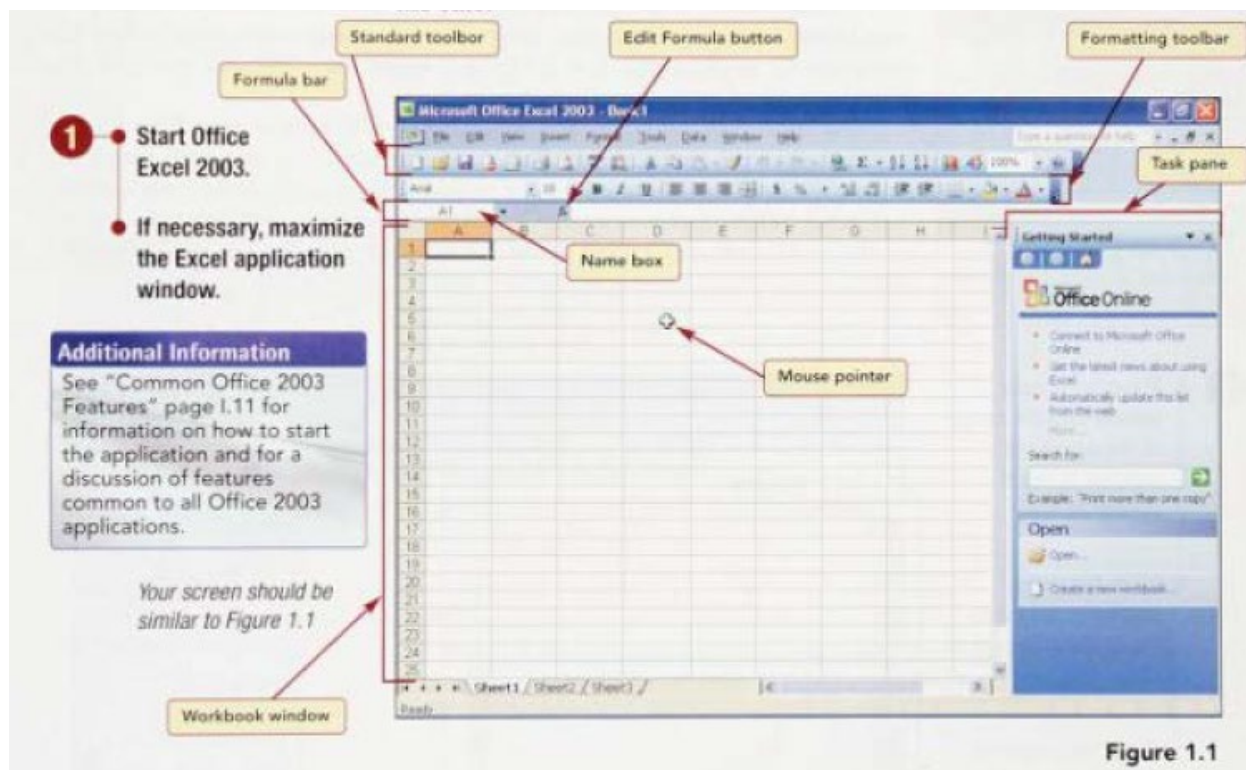
126. Panes are common interface components used to organize a layout. For example, an email client such as Microsoft Outlook has multiple panes. Here, panes may be used to display available folders, items in a selected folder, a selected item, and upcoming calendar events.

127. Indeed, Microsoft Suite such as Word 2003 used panes to organize the user interface layout for the user. Shown below, Word 2003 included a task pane allowing users to interact with certain features:



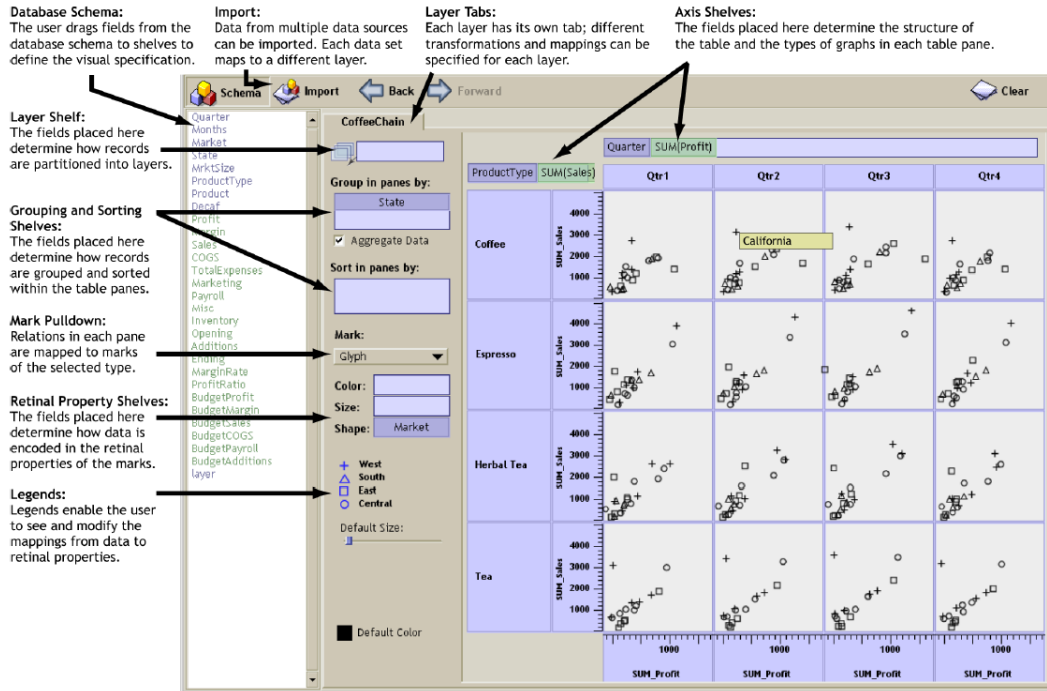
EX1011, 0034 (Figure 13) (annotated).

128. Excel 2003 also included panes such as a task pane:



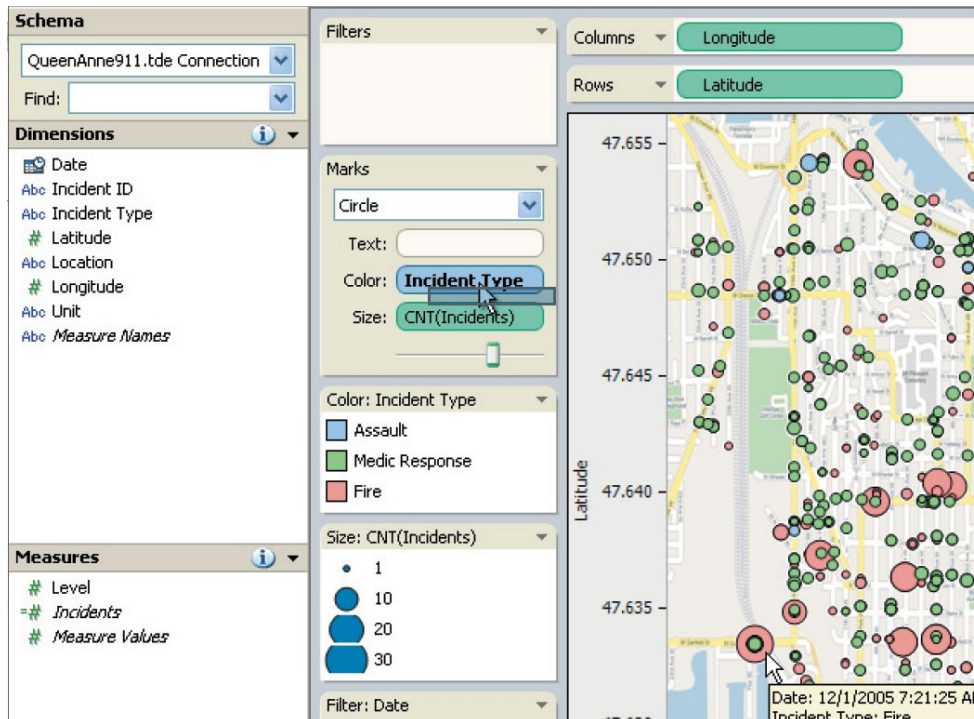
EX1011, 0044 (Figure 1.1).

129. Shown below, other data visualization tools such as Polaris also used panes to present data to users clearly:



EX1040, 0003.

130. Tableau also used panes:



EX1041, 0012; see also EX1049.

131. Thus, using panes to organize graphical user interfaces was also a widely known and used design choice well before 2008.

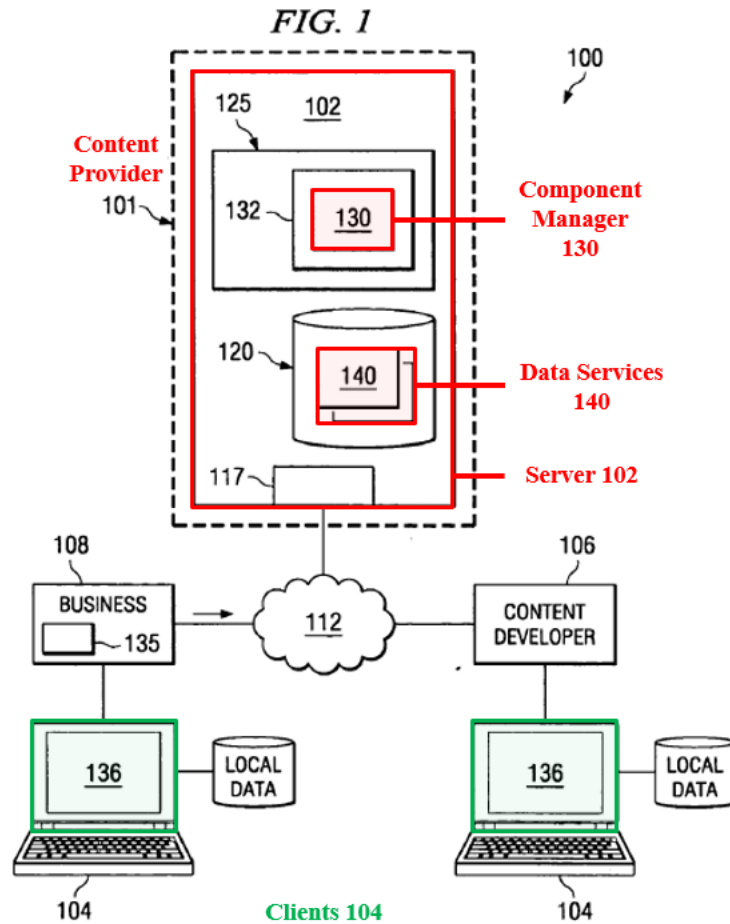
VII. Overview of the Prior Art

132. This section provides an overview of the prior art and demonstrates that publishing/embedding an interactive chart as a document was well known in the industry well before the priority date of the '000 patent.

A. Rostoker discloses a platform for generating, modifying, configuring or adapting interactive charts.

133. Rostoker describes “methods, systems, and software for providing a compilation of composite applications (‘xApps’) and xApp components that may be modified, reconfigured, and/or adapted to meet a user's desires or requirements of a business application.” EX1005, ¶4. “Each component or application includes one or more pre-constructed user interfaces (UIs) and performs one or more predetermined functions, services, or queries.” EX1005, ¶4.

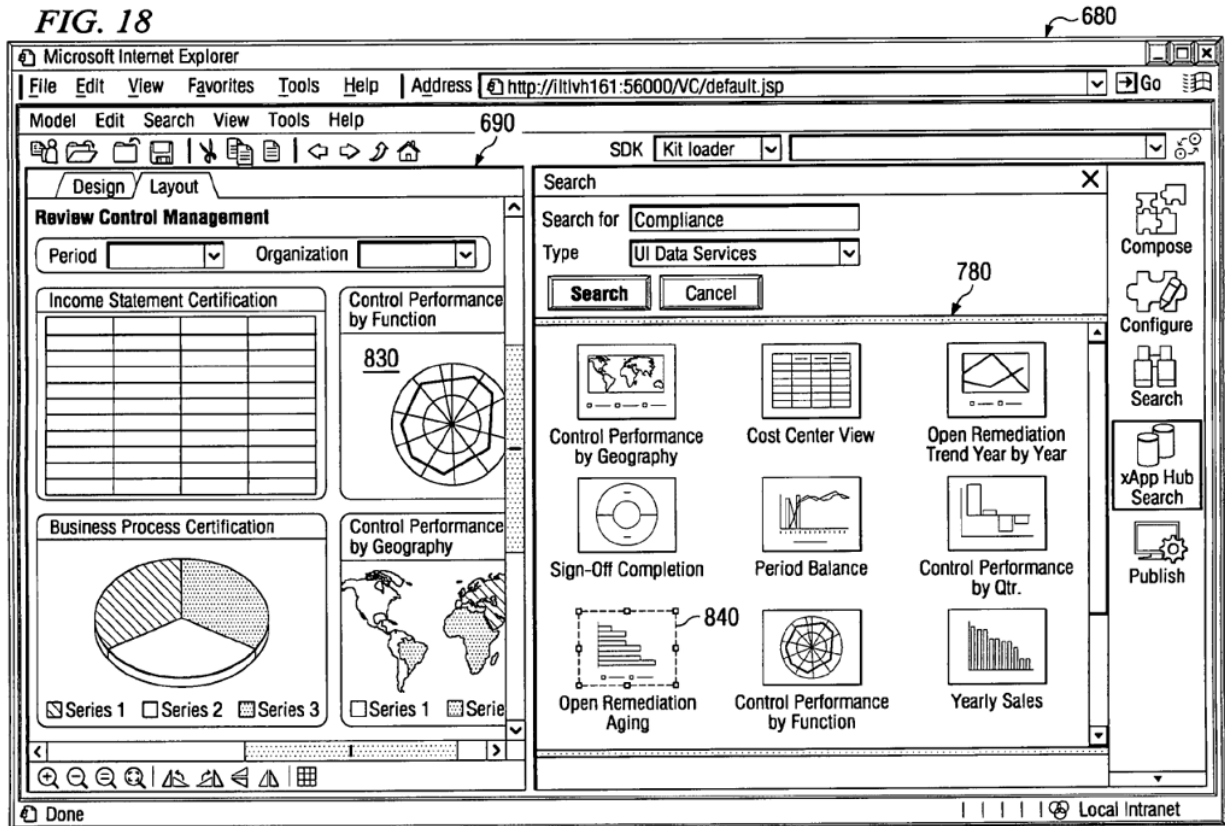
134. Figure 1 of Rostoker “illustrates an example system 100 for presenting, providing, or otherwise managing enterprise application content and other data services 140 via a content provider 101.” EX1005, ¶18. As illustrated below, “system 100” includes “server 102, one or more clients 104, one or more content developers (or vendors) 106, [and] one or more businesses 108, at least some of which communicate across network 112.” EX1005, ¶29.



EX1005, FIG. 1 (annotated).

135. Rostoker explains that “the content provider 101 uses a component manager 130, or other similar information hub of enterprise knowledge and components, to present a plurality of data services 140 to businesses 108, or their individual users, as well as other companies and users in the network.” EX1005, ¶19. And a “modeling environment 132 may be utilized to create, adapt, or configure the data services 140 (available to a user via the component manager

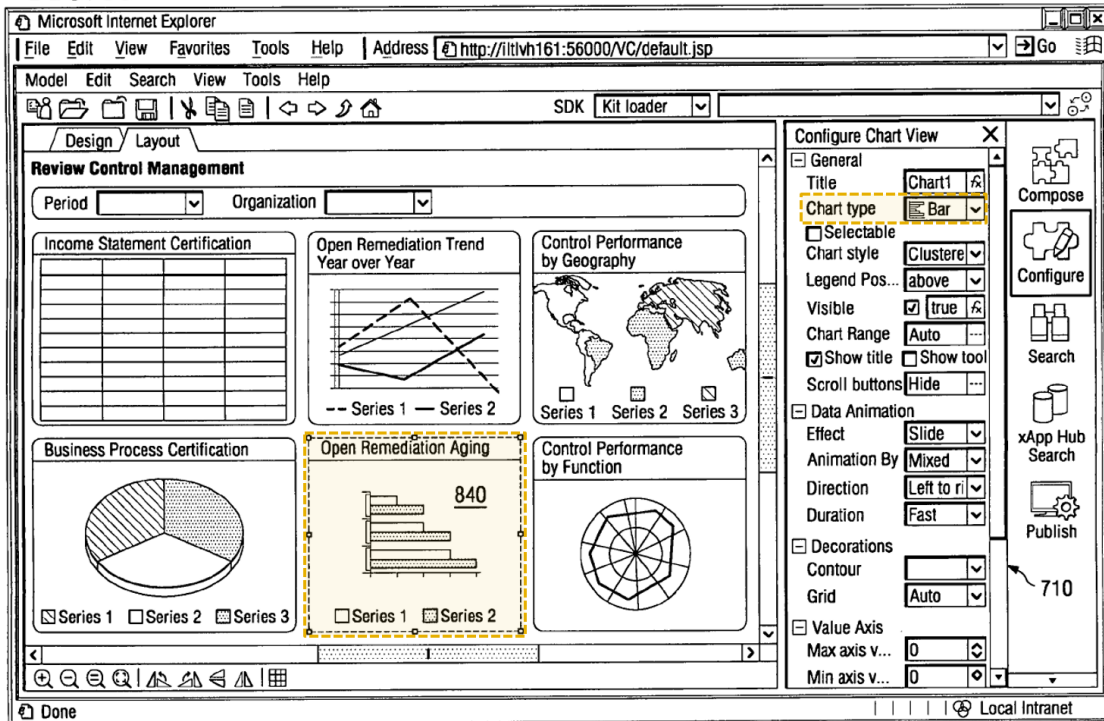
130) for creating a custom application, such as a business application, to fulfill a need of the user,” as illustrated, for example, in Figure 18. EX1005, ¶40.



EX1005, FIG. 18.

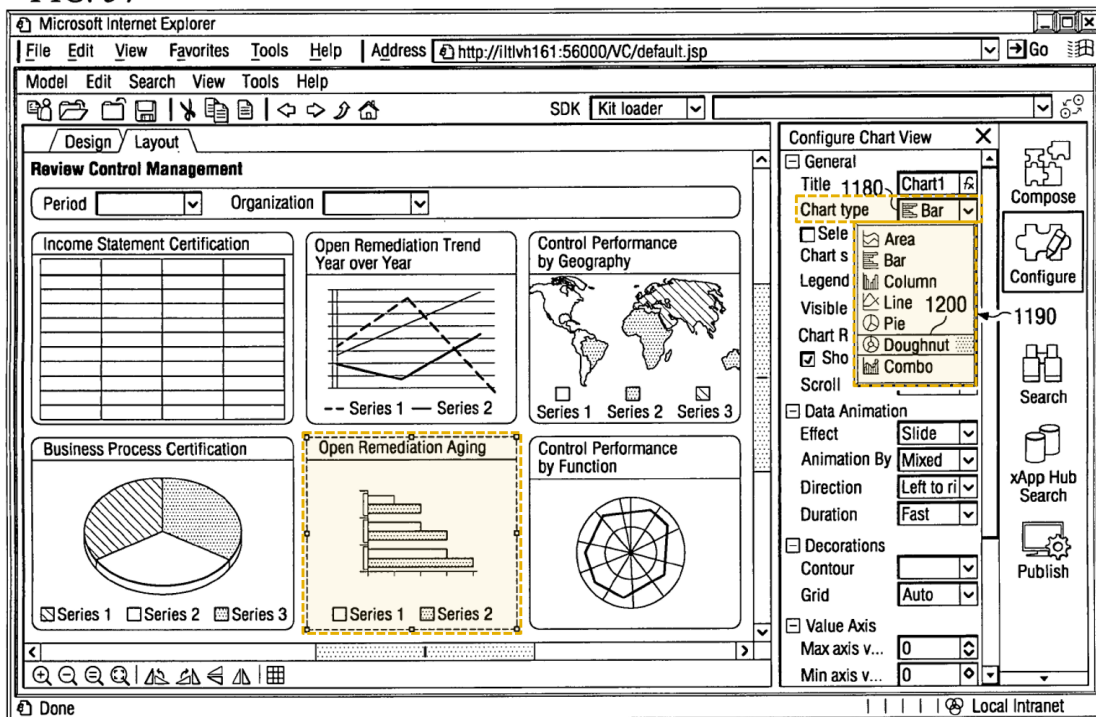
136. Figures 53-55 below show an example of configuring an xApp 840. Figure 53 shows the configure function for xApp 840 is selected, Figure 54 shows the user changes the chart type of the xApp 840 by selecting the desired type 1190 (e.g., “Doughnut” chart type 1200) from the list 1180, and Figure 55 shows the chart type of the xApp 840 is changed to the selected chart type 1200. EX1005, ¶71.

FIG. 53



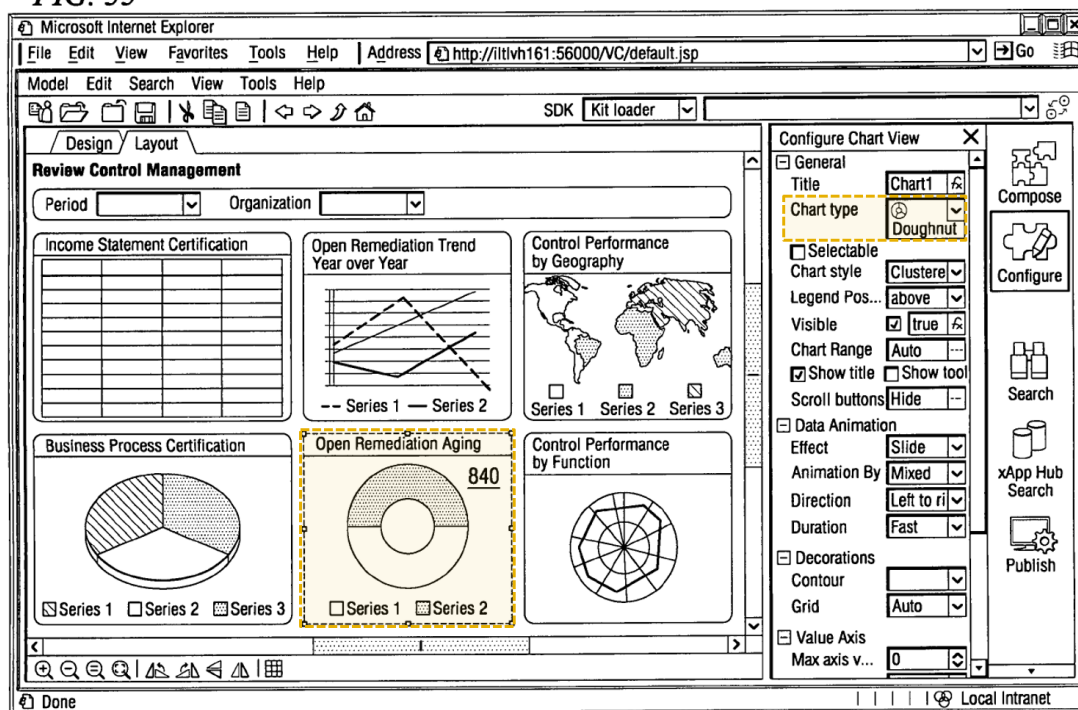
EX1005, FIG. 53 (annotated).

FIG. 54



EX1005, FIG. 54 (annotated).

FIG. 55

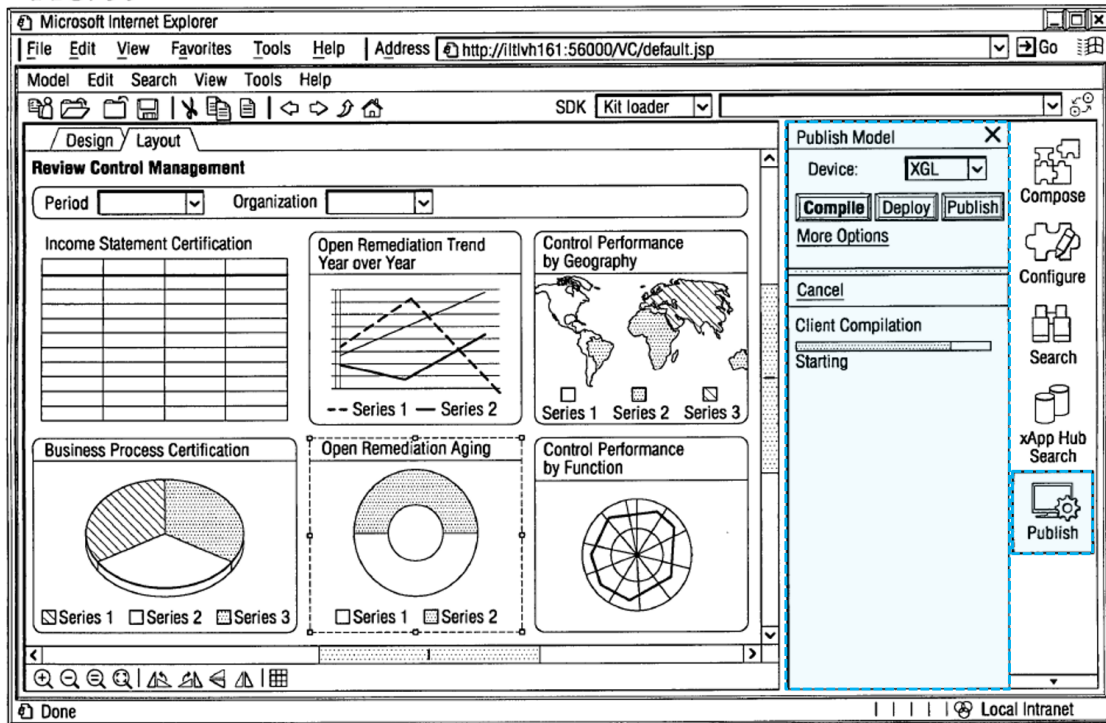


EX1005, FIG. 55 (annotated).

137. Rostoker also discloses that a user can “publish” the new or modified application by selecting “the Publish button 1210,” as shown in Figure 56 below.

EX1005, ¶71.

FIG. 56

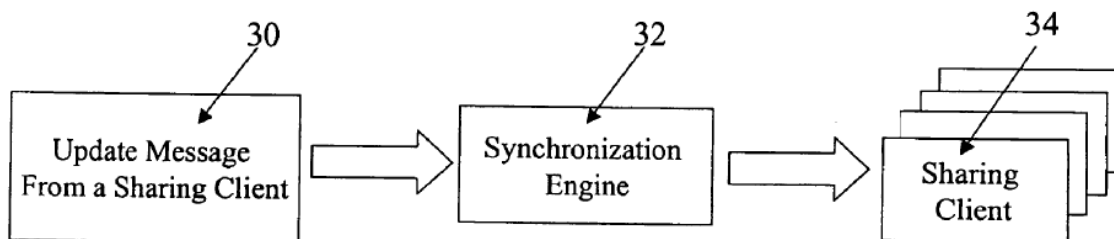


EX1005, FIG. 56 (annotated).

B. Strong discloses distributing a data object upon modification to the data object.

138. Strong is generally directed to “sharing files among multiple computing devices,” referred to as “clients” throughout Strong’s disclosure.

EX1067, ¶22. Figure 2 of Strong “depict[s] the basic paradigm for synchronizing a data object amongst the clients.” EX1067, ¶26.



EX1067, FIG. 2.

139. Strong explains that “changes or updates 30 from a sharing client 34 will be sent to the synchronization engine 32 which will forward or update a copy of a data object on each of the clients listed as part of the plurality of sharing clients 34 in order to synchronize that data object with other sharing clients.” EX1067, ¶26. For example, “if changes are made by one of the sharing clients 34 to a particular data object, an update message 30 is sent from the particular sharing client to the synchronization engine 32.” EX1067, ¶27. Strong further explains that “[t]he synchronization engine, in turn, will make the particular adjustment to the original or copy of the data object on the server and distribute the data object out to each of the sharing clients 34.” EX1067, ¶27. “[I]n some embodiments, updates are represented by sending a new copy of the entire object.” EX1067, ¶27; *see also* EX1067, ¶28 (“[W]hen an autosave occurs to the data or file being edited,” “[t]he autosaved version may be updated to the server having the synchronization engine and then pushed out or cascaded to the sharing clients.”). Thus, Strong discloses distributing a data object to sharing clients when a modification to the data object is made.

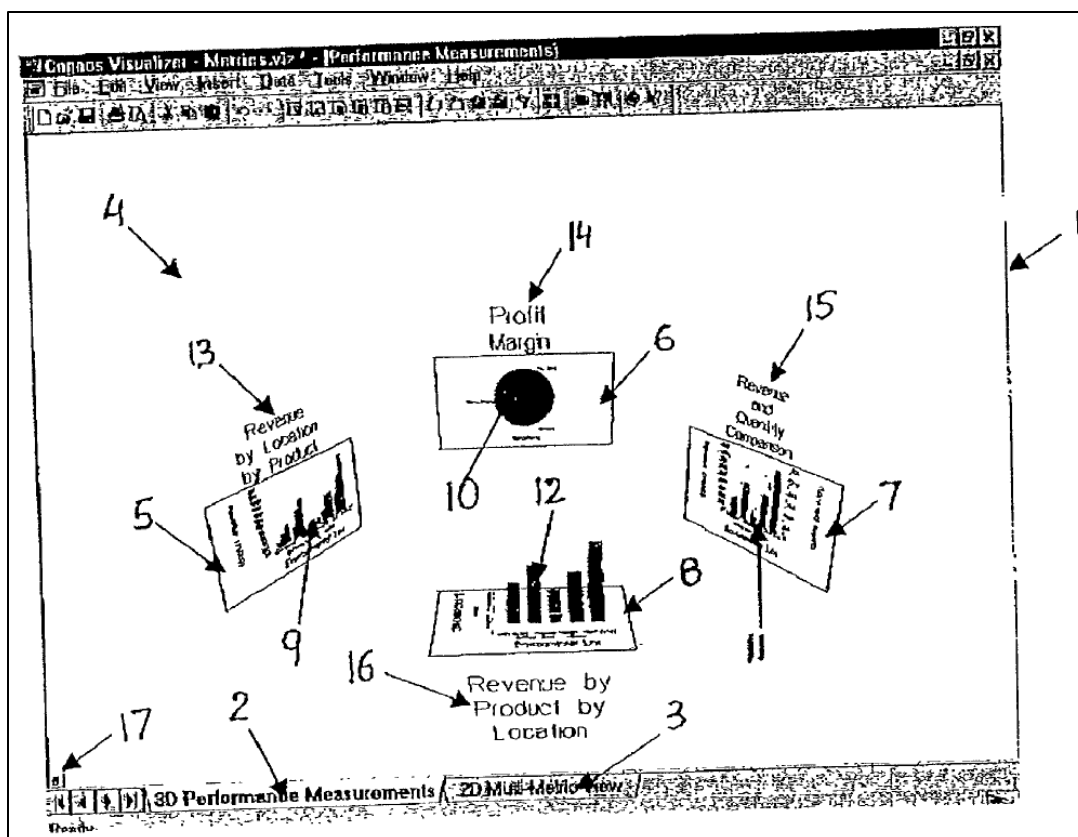
C. Jou discloses a visualization system for generating interactive charts.

140. Jou “provides an interactive system for visualizing business data.” EX1007, ¶10. Jou recognized that “the vast amounts of warehoused or otherwise collected business data would be useless without software tools for its analysis.”

EX1007, ¶8. “Business Intelligence (BI) applications” such as Cognos (mentioned above), provide “database querying, navigation and exploration tools . . . reporting, modeling, and visualization tools.” EX1007, ¶8. However Jou sought to improve these systems with “a new interactive system and method of data visualization, having many features lacking from prior art visualization tools.” EX1007, ¶9.

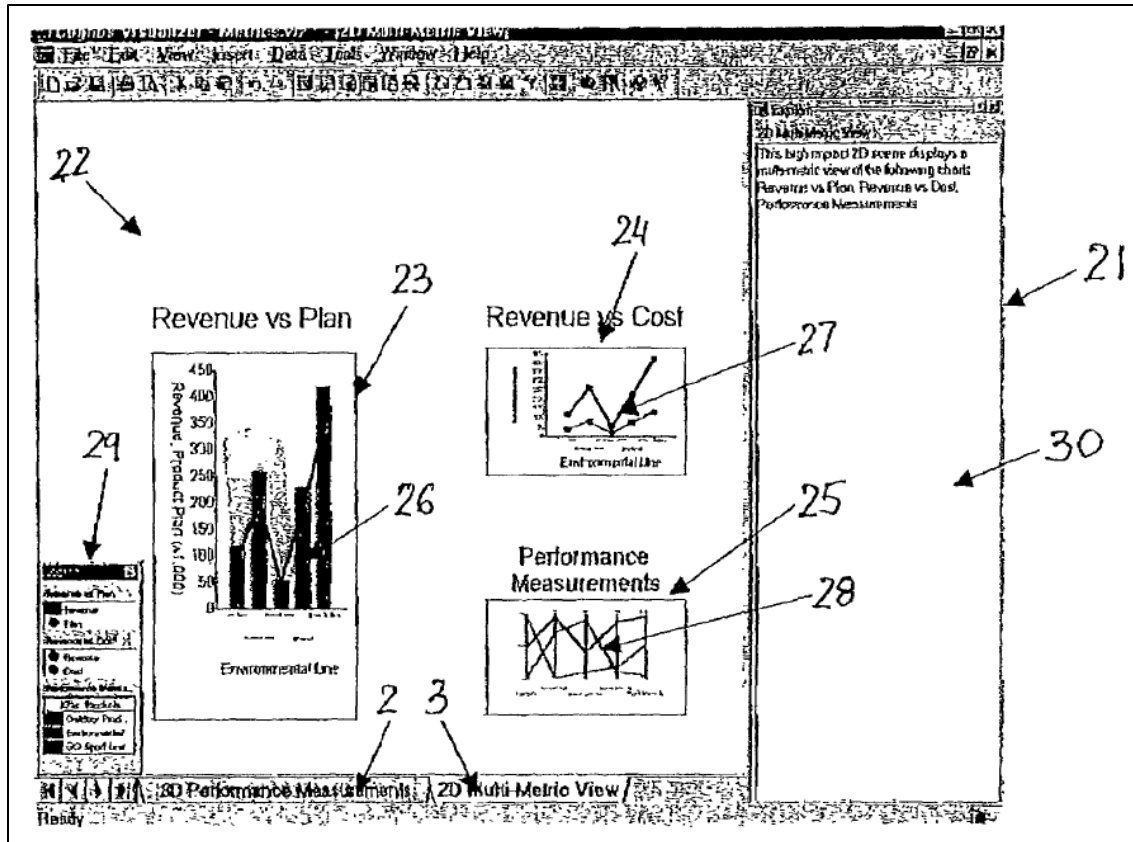
141. “A visualization . . . can be built . . . from a blank visualization and adding manually various visualization elements” EX1007, ¶13. Jou provides “a large variety of predefined types of two- and three-dimensional charts, such as line, bar, surface, pie, swatch, and map charts.” EX1007, ¶¶14, 36.

142. For example, Figure 2A below shows a visualization including charts 9-12. EX1007, ¶36. As shown, “chart 9 is a two-dimensional clustered bar chart, chart 10 is a two-dimensional pie chart, chart 11 is a two-dimensional bar chart combined with a two-dimensional line chart, and chart 12 is a three-dimensional bar chart.” EX1007, ¶36.



EX1007, FIG. 2A.

143. Figure 2B below shows a second sheet of the visualization including “a two-dimensional bar chart combined with a line chart (chart 26), a multi-line chart 27 and a parallel coordinate chart 28.” EX1007, ¶37.



EX1007, FIG. 2B.

144. Jou's charts include multiple "interactive features." For example, Jou discloses various "filter[s]," that only show a subset of data. EX1007, ¶38. As disclosed in Jou, this is beneficial to identify trends or patterns within the data. EX1007, ¶38. For instance, Jou discloses check, radio, and range filters allowing a user to select data visible in a chart. EX1007, ¶¶38-41. Jou also discloses an animation filter that shows data progression over time. EX1007, ¶42. For example, a three-dimensional bar chart may show sales by product over time. EX1007, ¶42. The animation filter may be used to animate the sales sequence as it changes "in a month-by-month sequence for a given year." EX1007, ¶42.

145. Jou also discloses “a drill-down feature,” “allow[ing] drilling on the dimension axis of charts, by moving either up or down the hierarchy of categories of a dimension . . . to obtain a more . . . detailed view of the data.” EX1007, ¶44. “[T]his is achieved by double-clicking an axis label, with the direction of the drilling indicated by the form of the cursor.” EX1007, ¶44. Jou further discloses a “zoom feature” where “the user can obtain a closer area view of a map or layer (zooming in), [or] obtain a wider area view of a map or layer (zooming out)” EX1007, ¶49.

D. Clark discloses a system for publishing charts to an enterprise website.

146. Clark is directed to “business data visualization and publishing systems.” EX1010, ¶3. Clark provides a system “that generates rich charts, graphs, and maps for enhanced visualization,” allowing users “to build web sites, portals, and dashboard[s] using a drag-n-drop interface.” EX1010, ¶¶28-29. Clark’s “data visualization and publishing system 100 provides a streamlined and intuitive interface for users with ordinary browsers to rapidly access data sources, visualize information, and publish results.” EX1010, ¶41. For example, “[t]o generate attractive charts and map graphics, The B3 system 200 can use the POPCHART and Optimap products sold by Corda.” EX1010, ¶52; *see also* EX1010, ¶47 (“POPCHART is described by Corda as a server-based suite of tools for creating sophisticated interactive charts and graphs in Macromedia® FLASH, SVG, PNG,

JPEG, PDF, EPS, TIFF, WBMP, etc.’). Therefore, Clark provides a “web-based” interface for “creating and publishing business dashboards.” EX1010, ¶52.

147. Clark further discloses that its system “can publish standalone dashboards, portals, or web pages, as well as components to be integrated into groupware or enterprise portals.” EX1010, ¶49; *see also* EX1010, ¶117 (“Publishing—create web pages, dashboards, and portals, or publish content to groupware and external portals”). Clark describes this “enterprise portal,” which resides, for example, “on [a] company’s intranet,” separate from the “web-based” interface for “creating and publishing business dashboards.” EX1010, ¶¶52, 137 (“[A dashboard] could also be published on the company’s intranet to enhance collaboration between employees”).

E. The QlikView Manuals disclose a visual analytics platform to create and publish interactive charts.

148. The QlikView Reference Manual, v8.50, 1st ed. (EX1008) and the QlikView Server Reference Manual, V8.50, 1st ed. (EX1009) (collectively “QlikView Manuals”) are for Version 8.50 of QlikView, a well-known data visualization tool. EX1028, 0002, 0017-18.

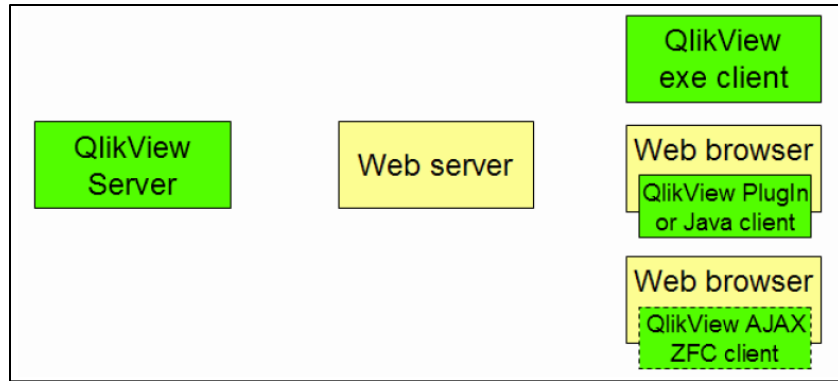
149. In early 2008, a POSA would have been aware of QlikView and the QlikView Manuals. The QlikView Manuals describe how QlikView works.

150. By at least June 26, 2008, QlikTech announced that QlikView version 8.50 was scheduled for release on June 30, 2008. EX1022, 0001; EX1068, ¶7. A

POSA would have understood that the manuals would have been in the QlikView 8.50 installation package when sold. *See* EX1014; EX1015; EX1016; EX1017. Additionally, a POSA would have understood that a call to QlikTech's customer support would have allowed her to ask about and receive the QlikView Manuals. EX1020, 0001. From my experience, in 2008, companies such as QlikTech were eager to share how their products worked to gain users and market share.

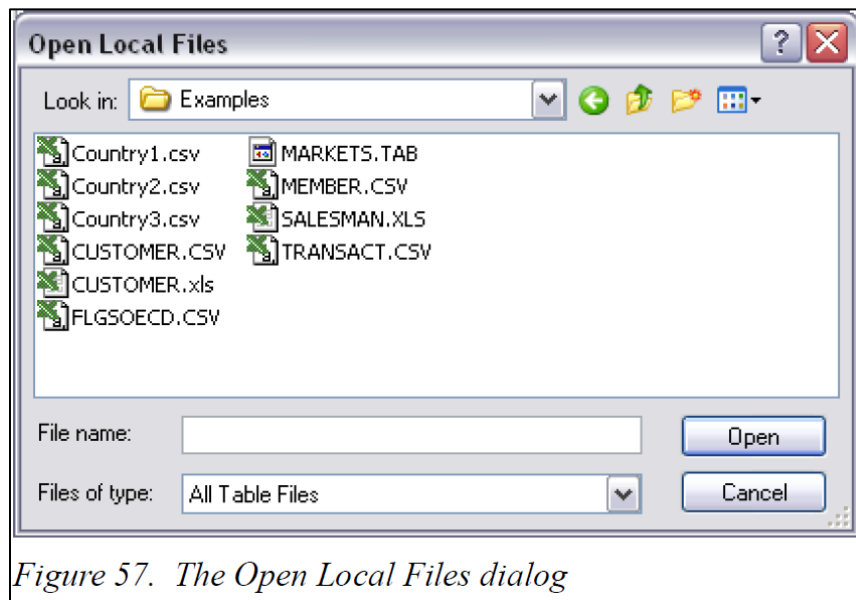
151. I have not seen any evidence showing QlikTech restricting access to the QlikView Manuals. To the contrary, evidence shows QlikTech distributing the QlikView Manuals to anyone who wanted to see them. In addition to including the QlikView Manuals in the installation package, QlikTech published the QlikView Manuals online, available for download. EX1068, ¶7. By at least July 8, 2008, QlikView included links to the installation package for Version 8.5 of QlikView. EX1015. From this link, a POSA would have known how find the QlikView Manuals on QlikTech's ftp website. *See* EX1014; EX1015; EX1016; EX1017.

152. The QlikView Manuals describe Version 8.50 of QlikView. Shown below, QlikView was a computer system having a QlikView Server, a Web server, and clients. A Qlik client can be a Java Client, Java Objects Client, Windows Client, Internet Explorer Client, QlikX Client, or an AJAX Client EX1009, 0123.



EX1009, 0095.

153. With QlikView, a developer loads data from databases, tables (e.g., Excel), web files, XML, or other QlikView files. EX1008, 0242-251. For example, as shown below the user can select and load a file stored locally.



EX1008, 0249.

154. Once data was loaded, the developer generates interactive charts using a Chart Wizard. EX1008, 0881-0947. Shown below, the user can use the Chart Wizard to select a chart type such as a bar chart, line chart, combo chart, radar

chart, scatter chart, grid chart, pie chart, funnel chart, block chart, gauge chart, pivot table, or straight table. EX1008, 0883-85. The Chart Wizard is also used to specify and customize the chart's title, determine whether the user can interact with the chart (e.g., read only), increase memory allocated to the chart, and define custom error messages. EX1008, 0882-86.

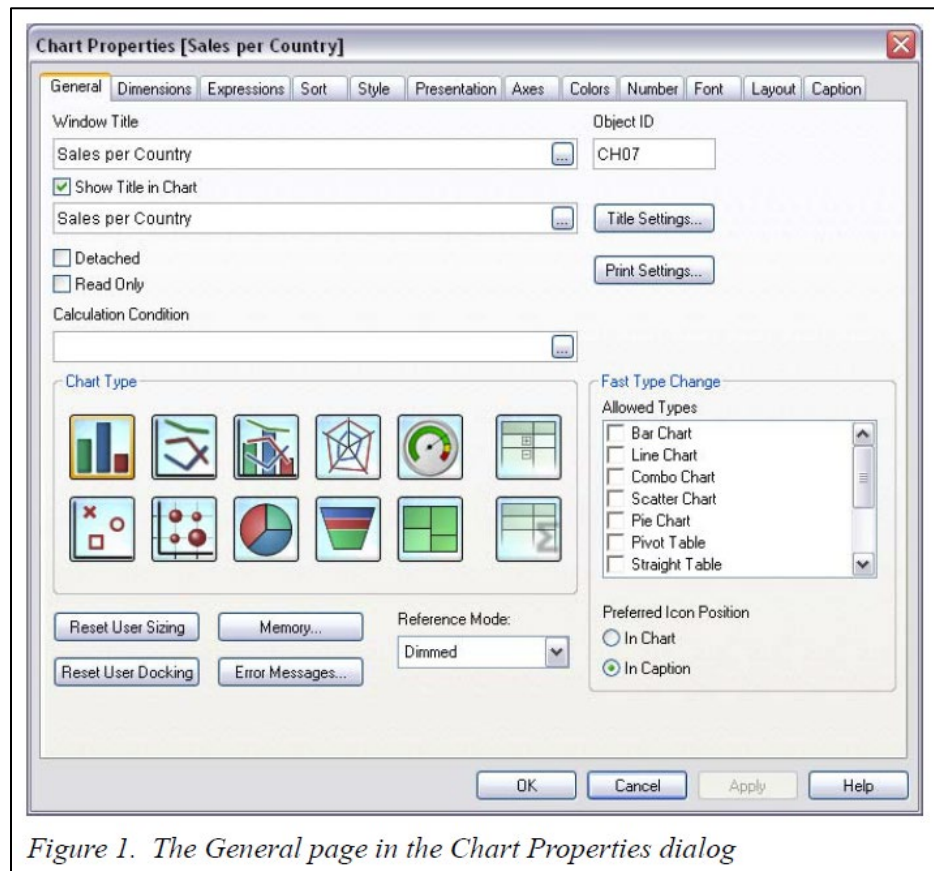
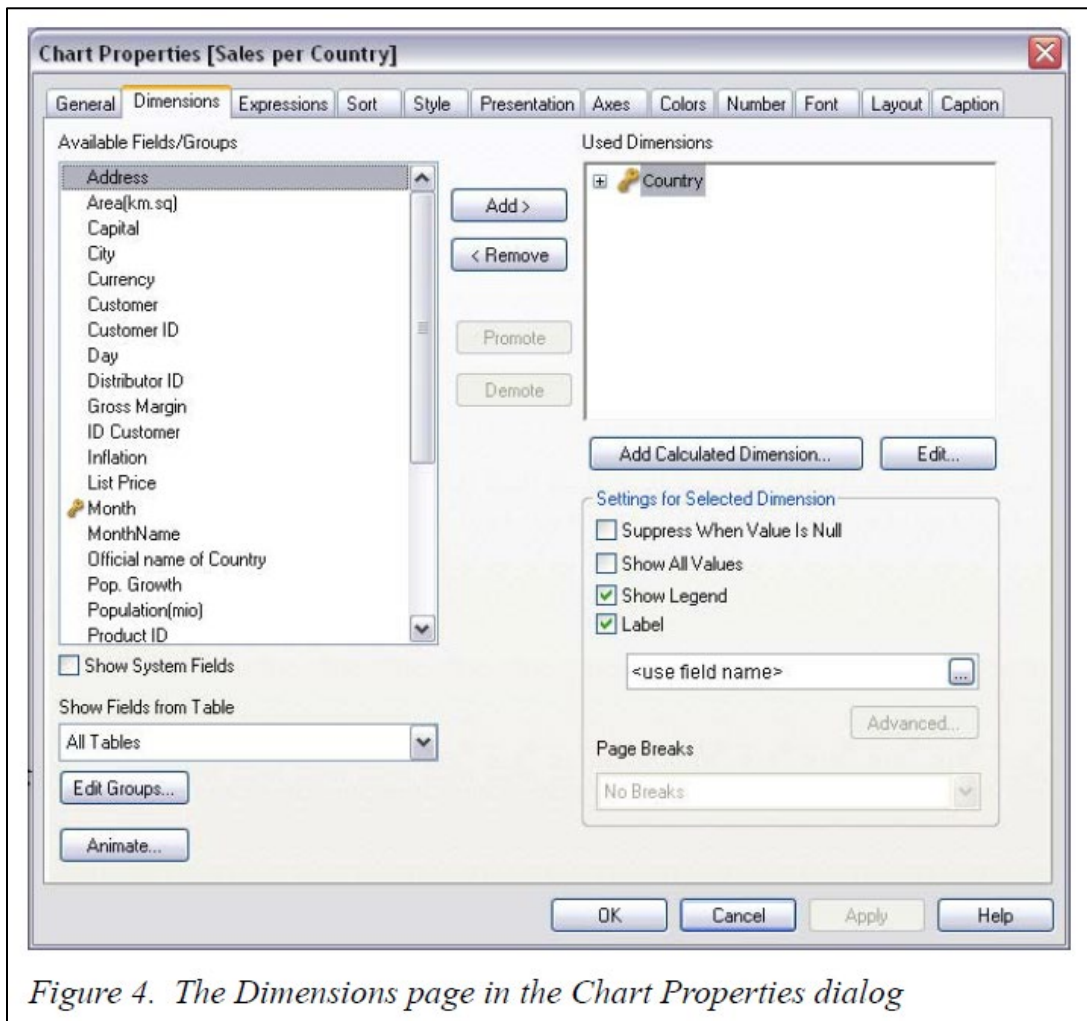


Figure 1. The General page in the Chart Properties dialog

EX1008, 0882.

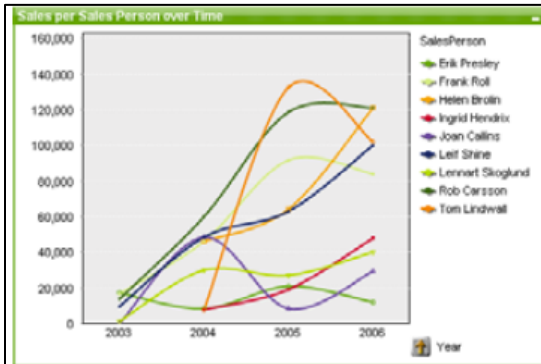
155. Once the chart type is selected, numerous chart properties can be customized. For example, the user can set the chart dimension (e.g., x-axis data type), animations, expressions to calculate chart content (e.g., y-axis data), how to

sort data, chart style (e.g., themes, orientation), presentation settings (e.g., enabling mouse over pop-up labels), legend settings, reference lines, chart text, chart axes, chart colors, number formatting (e.g., integer, date, time), font styles, layout settings (e.g., chart border), and chart captions. EX1008, 0894-0947. As an example, the dialog used to customize a chart's dimensions is shown below.

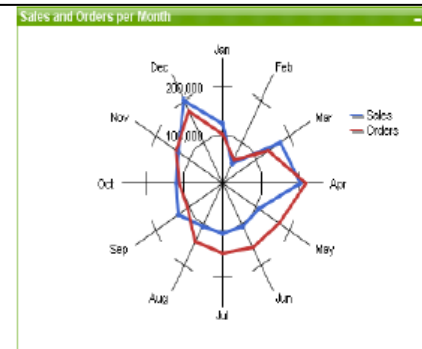


EX1008, 0894.

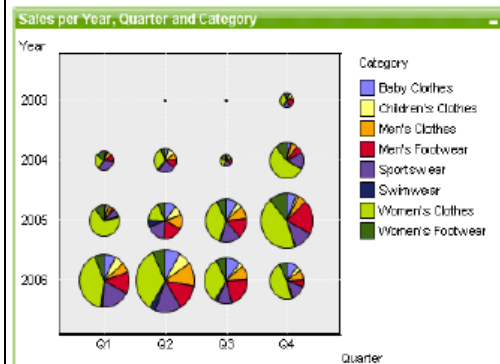
156. Below are example charts created by QlikView.



Line chart



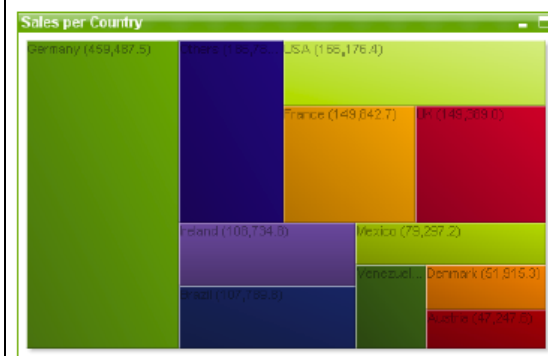
Radar chart



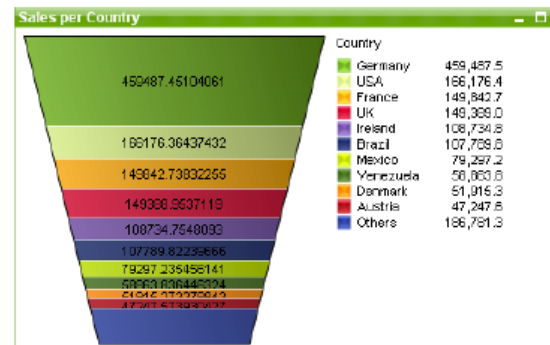
Grid chart



Gauge chart



Block chart



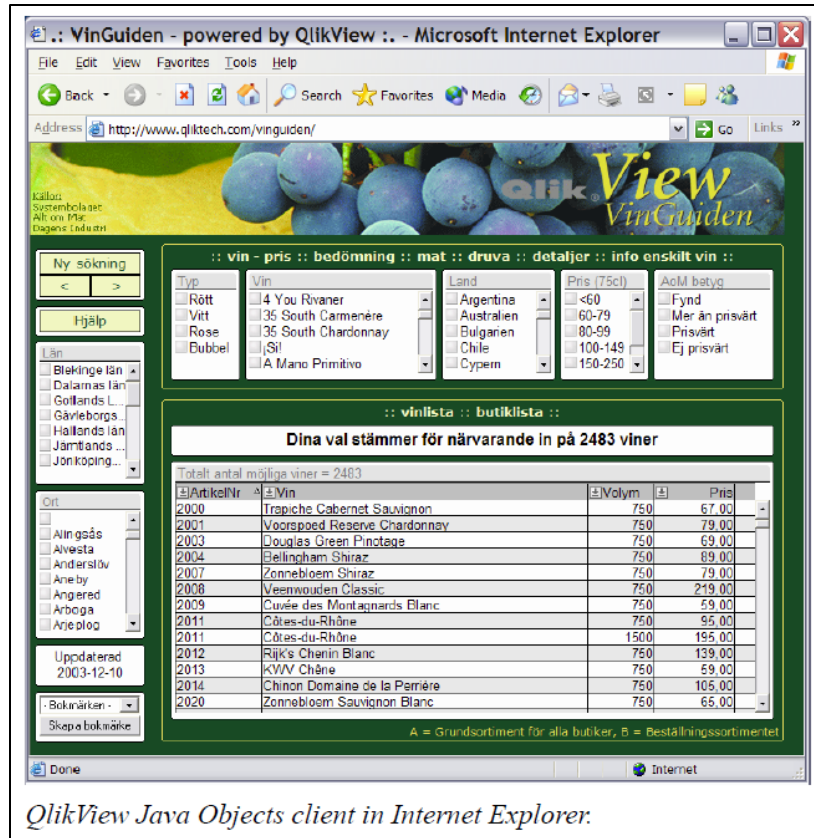
Funnel chart

EX1008, 0888.

157. QlikView's Chart Wizard provided many customizable options, including interactive features such as “drop-down,” “slider object[s],” “buttons,”

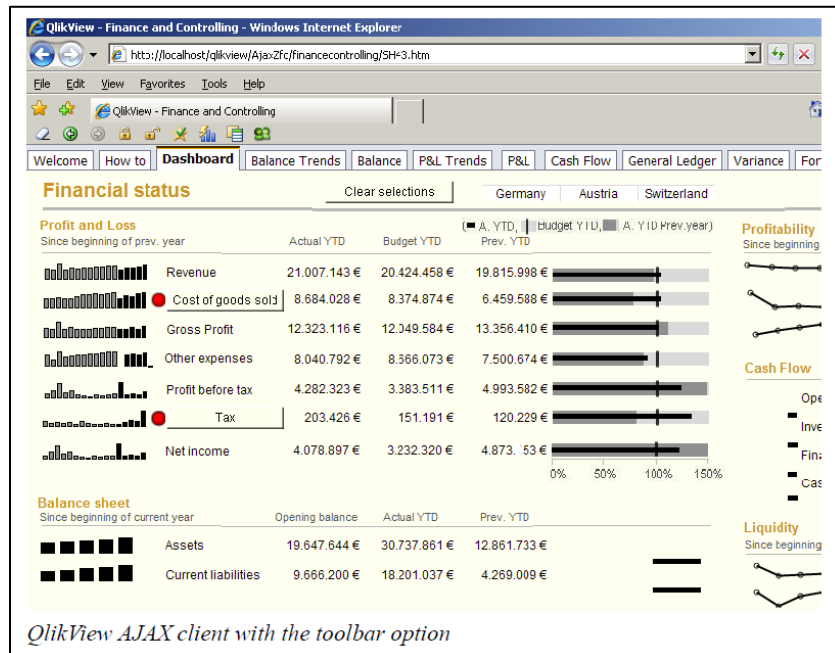
“interactive search,” “drill-down” “cyclic groups,” and “pop-up labels.” EX1009, 0016-17; EX1008, 0035-36, 0583, 0919, 1306-07. For example, when customizing a chart’s presentation settings, pop-up labels can show dimension values and expressions when the user moves the cursor over a chart data point. EX1008, 0919. Additionally, when hierarchical data fields are displayed in a chart, the user can drill-down into the field, to show the next field within the hierarchy. EX1008, 1306. For example, if the data in a chart has a hierarchy of: Year, Quarter, Month, then Year will be displayed as the chart dimension (e.g., axis). EX1008, 1306. However, if a single year is clicked, the chart dimension will update to show each Quarter for the selected year. EX1008, 1306.

158. Once generated, a developer can publish or embed the interactive chart as a webpage, accessible to other users via a web client (e.g., Java Client, Java Objects Client, Windows Client, Internet Explorer Client, QlikX Client, and AJAX Client) shown below. EX1009, 0123.



QlikView Java Objects client in Internet Explorer.

EX1009, 0133.



QlikView AJAX client with the toolbar option

EX1009, 0155.

159. The QlikView charts remain interactive. For example, they include “drop-down,” “slider object[s],” “buttons,” “interactive search,” etc. EX1009, 0016-17; EX1008, 0035-36. Column sizes can be changed, and input fields can be edited. EX1009, 0019, 0057. For example, charts published using the AJAX client included support for interactive features such as drill-down, drop-down select, sort, and selection indicators (e.g., LEDs). EX1008, 0035-36.

F. Bales discloses drag and drop for updating web content.

160. Bales discloses a collaborative computing environment, e.g., a portal, that displays collaboration resources and provides “[d]rag and drop functionality” for a community member to drag drop some resource into a target area. EX1006, Abstract. A portal is a gateway or an access point for users to various types of content and other collections of information. EX1006, ¶20. Bales provides a community framework for introducing a sense of memberships and communities that can be implemented on a portal application server. EX1006, ¶52.

161. Bales explains that a community “may be composed of various collaboration resources” and “community services.” EX1006, ¶55. Collaboration resources are “various content data types” and are stored in a repository. EX1006, ¶57. “Some examples of collaboration resources include documents, issues, discussions, links, events, announcements, as well as other types of files, data and

services.” EX1006, ¶57. “Community services [] are individual portlets that provide various services and functionality to the community.” EX1006, ¶60.

162. The “drag and drop functionality” for its community services allows a user to “drag and drop various resources and portlets from one page element to another.” EX1006, ¶158. Specifically, Bales discloses a “highly user-friendly and fluid interface” to enable the drag and drop feature. EX1006, ¶161. For example, Bales discloses a tag called “resourceDropZone” that “identifies an area on the page where draggable resources can be dropped.” EX1006, ¶165.

163. Figure 16 below illustrates an example of the drag and drop feature. EX1006, ¶159. In this example, the user is shown as dragging another resource named “Sample Groupnote” (1602) and dropping it into the related content field (1604) of the “Sample Issue” resource (1600). EX1006, ¶159.

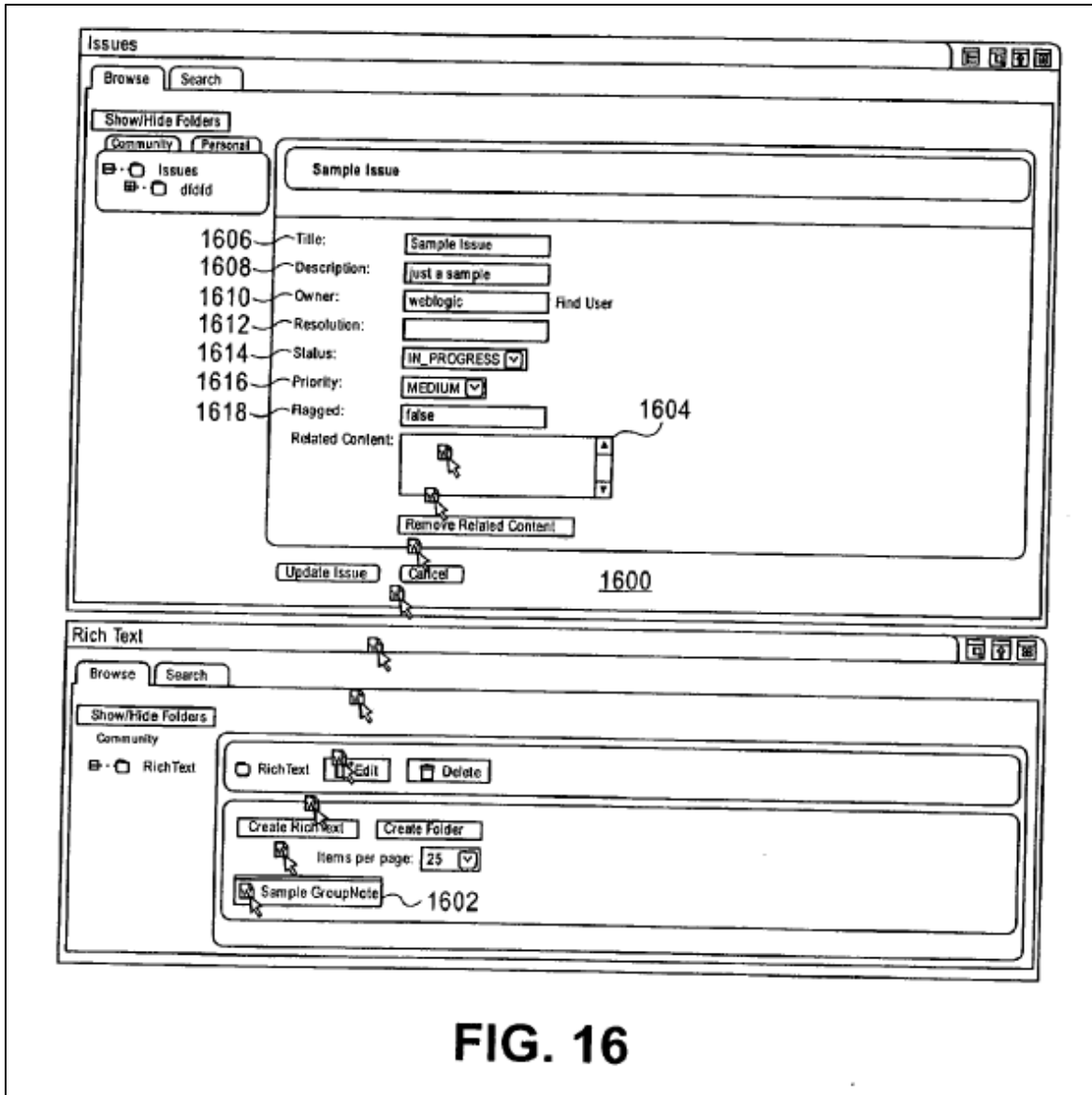


FIG. 16

EX1006, FIG. 16.

164. Accordingly, as shown in Figure 17 below, a link to the resource “Sample GroupNote” (1702) now appears in the related content (1704) field of the resource “Sample Issue” (1700) and a user can view the “Sample GroupNote” on his desktop. EX1006, ¶160.

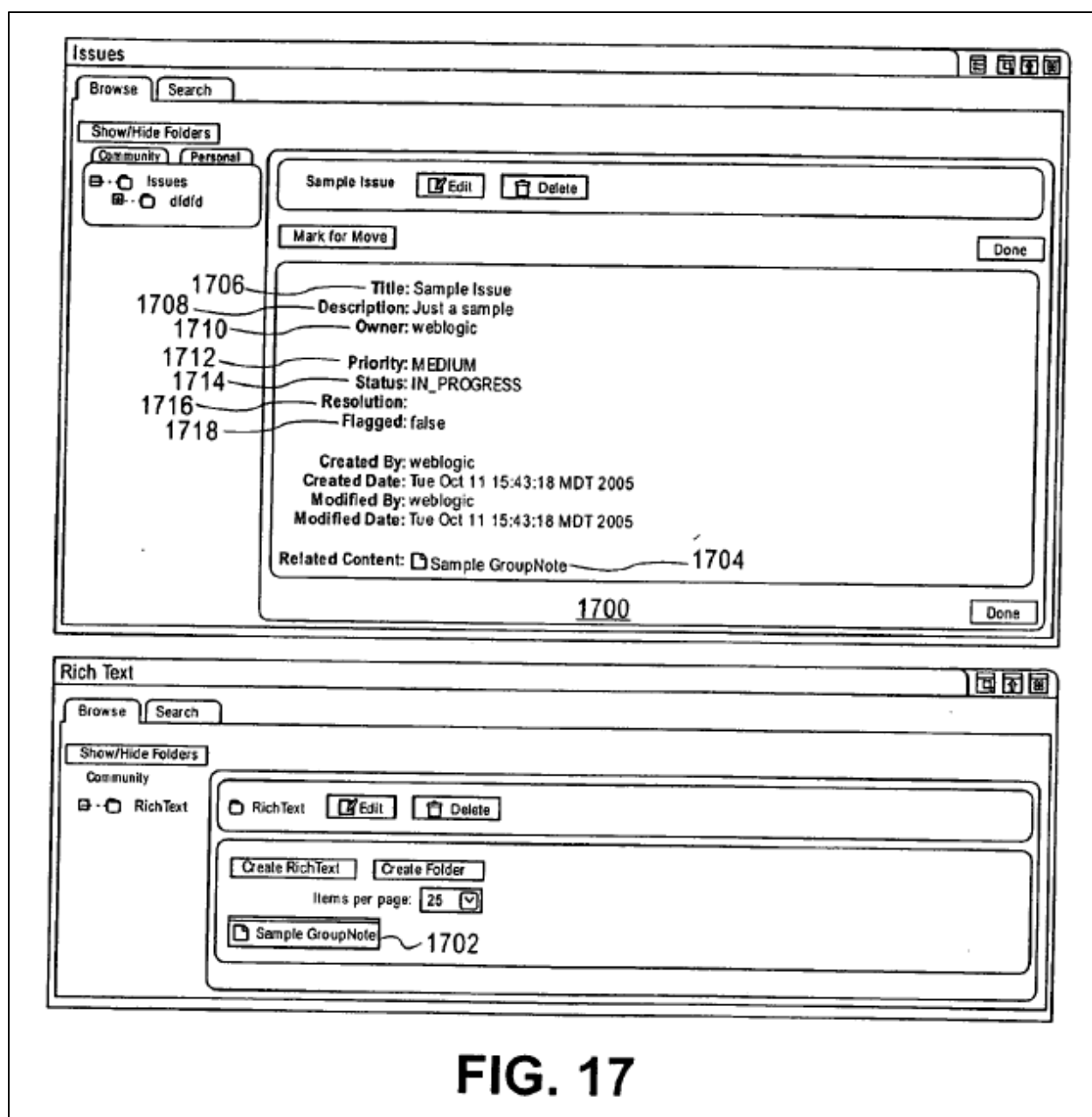


FIG. 17

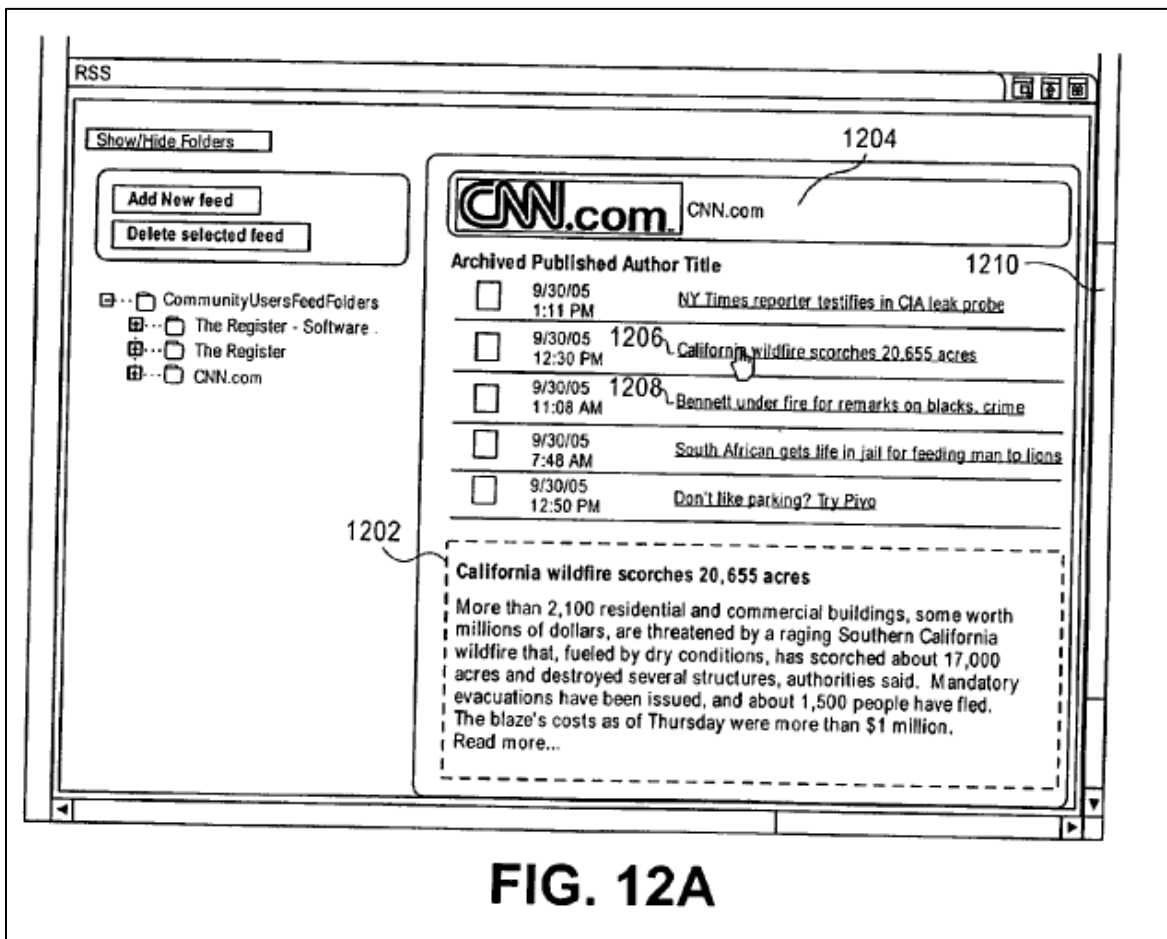
EX1006, FIG. 17.

165. Bales also discloses a dynamic content feature for the communities framework. EX1006, ¶122. “The dynamic content feature provides users with the ability to update a certain portion or portlet of a page or a portion of a portlet”

“without reloading the entire browser” each time an interaction occurs. EX1006, ¶122.

166. For example, Figure 12A below illustrates an example of a graphical user interface during the execution of dynamic content feature. EX1006, ¶126.

“The dynamic container (1202) is set to display a selected news story or a portion thereof.” EX1006, ¶126. When the user selects the news story “California wildfire scorches 20,655 acres” (1206), “the dynamic container (1202) is displaying the text of the appropriate news story.” EX1006, ¶126.



EX1006, FIG. 12A.

VIII. Grounds of Unpatentability

167. The three grounds of unpatentability in this declaration are listed below. I understand that all three grounds are based on obviousness. This declaration explains why the references in each ground renders obvious the claims.

Ground	References	Claims
1	Rostoker and Strong	1-4
2	Rostoker, Strong, and Jou	1-4
3	Rostoker, Strong, and Clark	1-4
4	QlikView Manuals and Bales	1, 3, 4

IX. Ground 1: Rostoker in view of Strong renders obvious claims 1-4.

168. Rostoker in view of Strong teaches or suggests every element in claims 1-4.

A. Independent Claim 1

1. [1P] “A computer-implemented method”

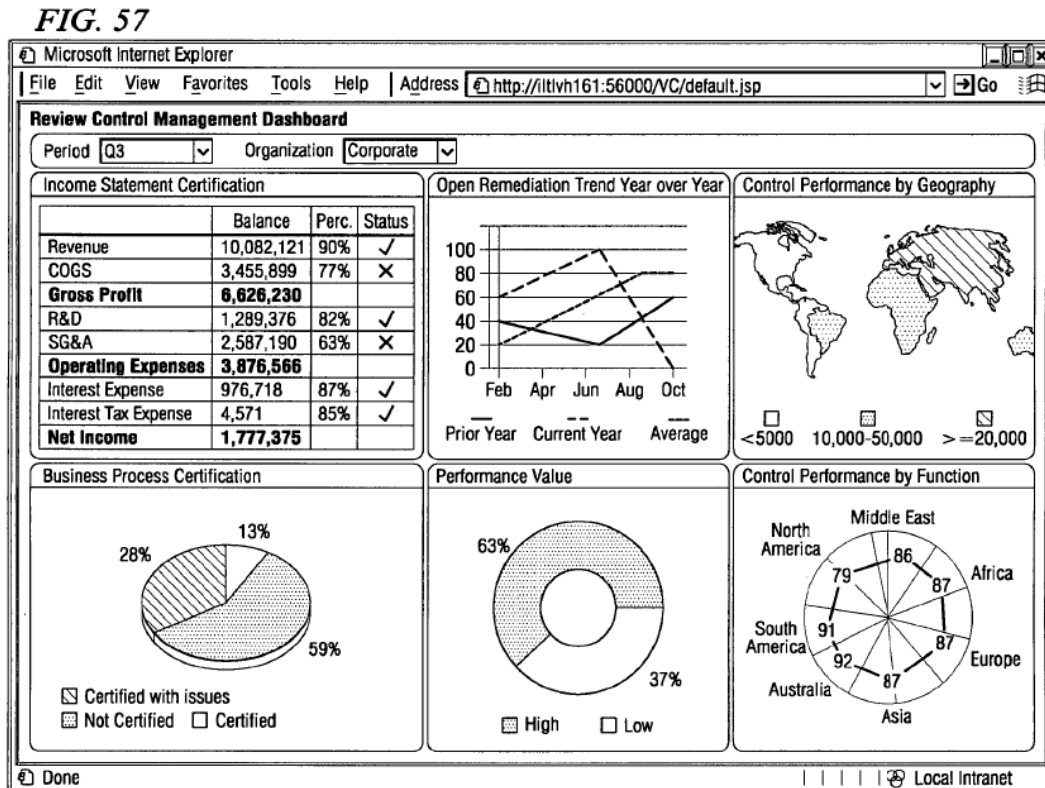
169. The preamble of claim 1 recites “[a] *computer-implemented method*.” I understand that in certain circumstances the preamble of a claim may be limiting. To the extent the preamble is determined to be limiting, it is my opinion that Rostoker teaches the preamble of claim 1.

170. As I discussed above, Rostoker is directed to “**methods**, systems, and software for providing a compilation of composite applications (‘xApps’) and xApp components that may be modified, reconfigured, and/or adapted to meet a user's desires or requirements of a business application.” EX1005, ¶4.¹ “Each component or application includes one or more pre-constructed user interfaces (UIs) and performs one or more predetermined functions, services, or queries.” EX1005, ¶4.

171. Rostoker discloses “a storefront having a knowledge base or compilation of software components, applications, or other data services that can easily be modified or integrated into different business analyst and end user tools.” EX1005, ¶6. “The user may locate the components by searching the hub,” and “[t]he user may then select a pre-existing application or component and use the selected component or application without modification or use the component or application as a template, making modifications in order to create an application to operate in a desired manner.” EX1005, ¶6. Rostoker also discloses that “[a] search may return one or more applications, application components, or other business process components that the user may import and configure to operate in a predetermined manner.” EX1005, ¶8.

¹ Emphasis added unless otherwise indicated.

172. Figure 57 of Rostoker illustrates an example application generated by Rostoker's methods that includes multiple chart-based "xApps." EX1005, ¶71. For example, as shown below, the application displayed in Figure 57 includes bar charts, maps, pie charts, and donut charts.



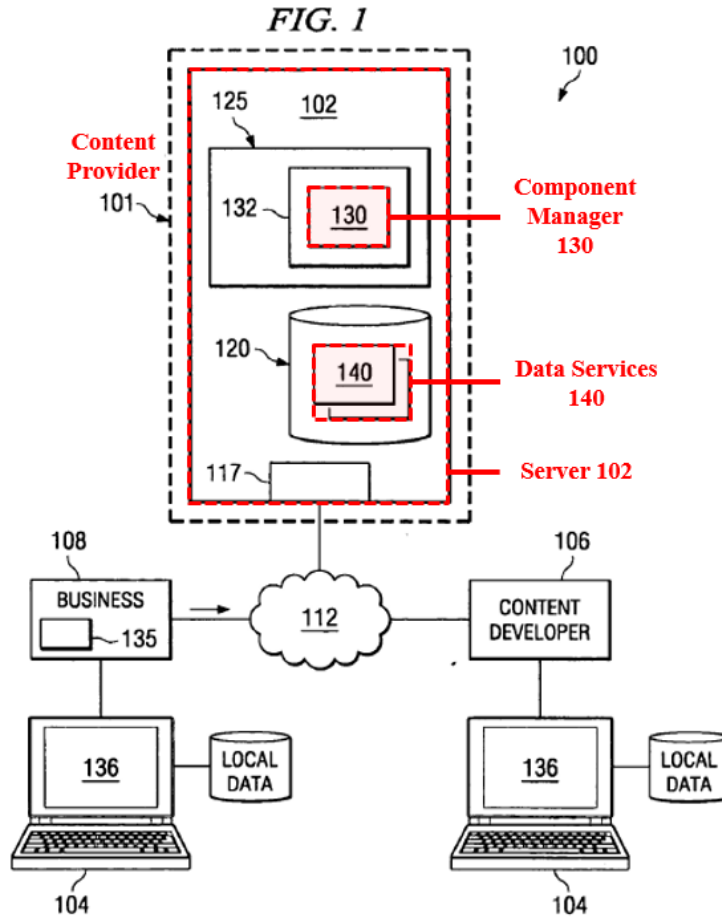
EX1005, FIG. 57.

173. Therefore, it is my opinion that Rostoker discloses the preamble of claim 1.

2. [1A] “receiving a selection, on a first website, of at least one interactive chart from a plurality of interactive charts”

174. Element [1A] recites “*receiving a selection, on a first website, of at least one interactive chart from a plurality of interactive charts.*” In my opinion, Rostoker teaches this limitation.

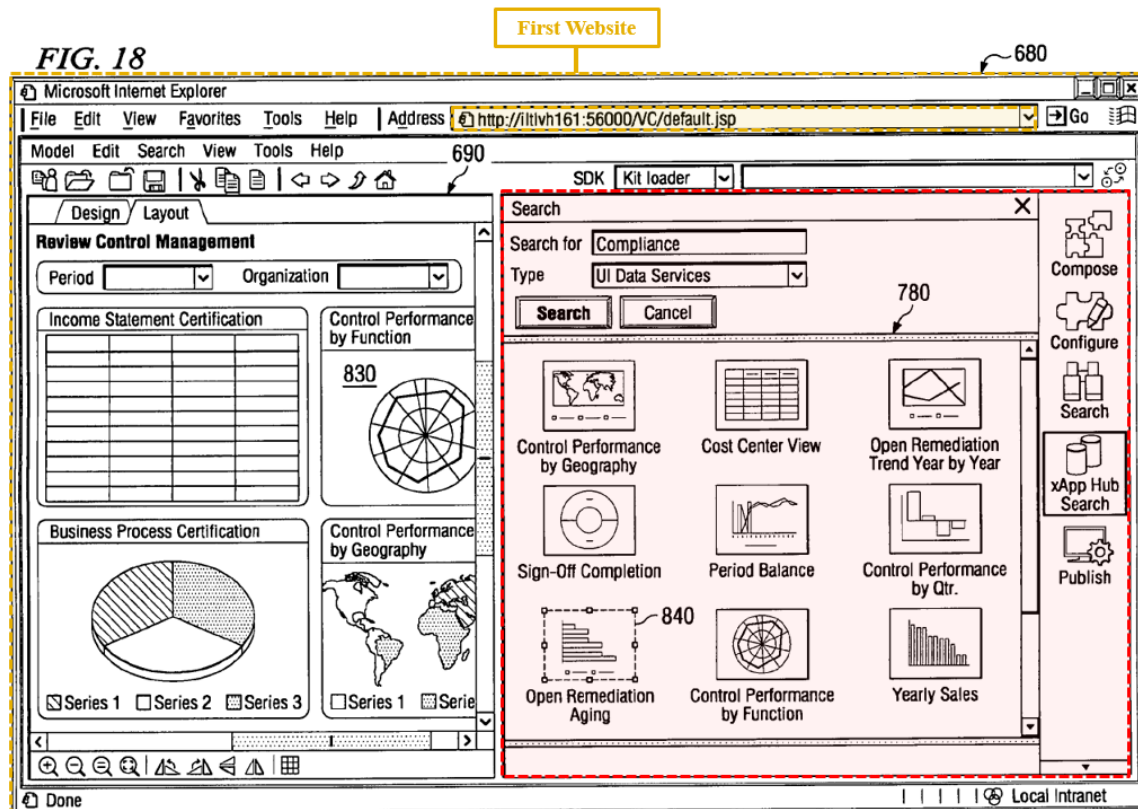
175. In my opinion, Rostoker discloses “*a first website,*” and “*a plurality of . . . charts*” on the first website. For example, Figure 1 of Rostoker “illustrates an example system 100 for presenting, providing, or otherwise managing enterprise application content and other data services 140 via a content provider 101.” EX1005, ¶18. As illustrated below, “system 100” includes “server 102, one or more clients 104, one or more content developers (or vendors) 106, [and] one or more businesses 108, at least some of which communicate across network 112.” EX1005, ¶29.



176. Rostoker’s system includes “a storefront having a knowledge base or compilation of software components, applications, or other data services that can easily be modified or integrated into different business analyst and end user tools.” EX1005, ¶6. Rostoker explains that “content provider 101 may be considered a storefront, hub, or retailer of business content and solutions.” EX1005, ¶18. Rostoker’s hub or “storefront can typically be accessed in a stand-alone fashion (such as a **website**).” EX1005, ¶18. Rostoker further discloses that “**server 102** [of content provider 101] may also include or be communicably coupled with a **web**

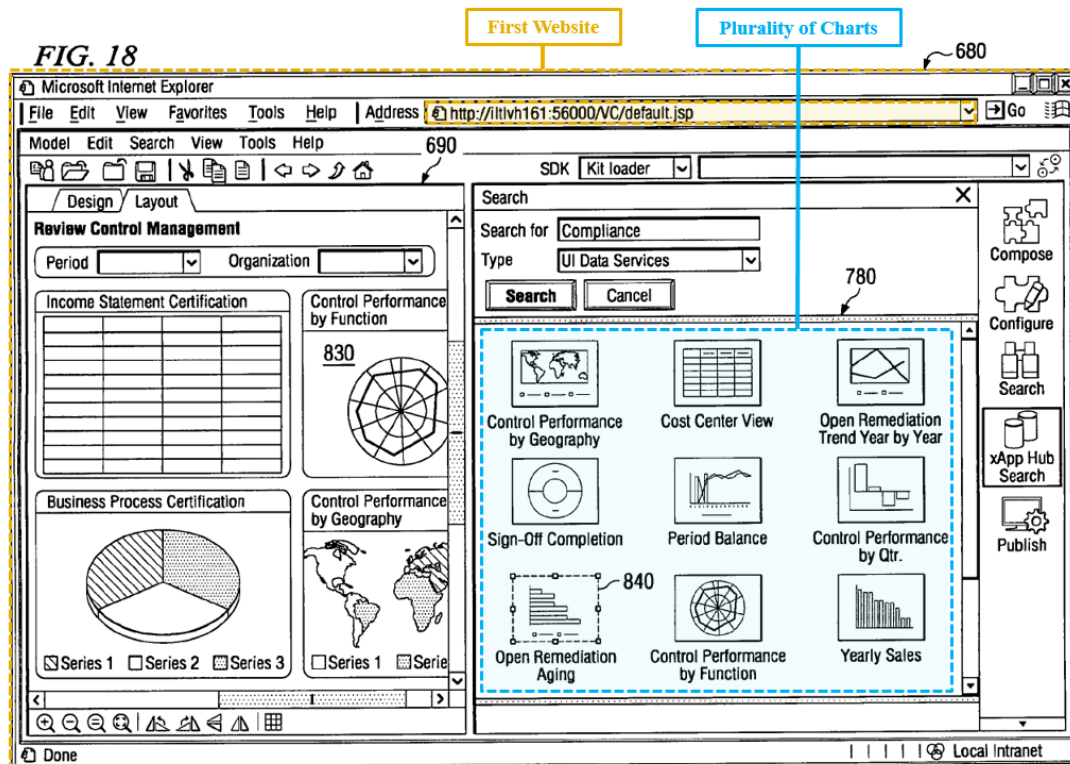
server” EX1005, ¶29. The “storefront” or “hub” is provided via “a component manager 130,” which “present[s] a plurality of data services 140 to businesses 108, or their individual users, as well as other companies and users in the network.” EX1005, ¶19. Rostoker discloses that “the content provider 101 may provide or utilize a bank of easily searched xApps and xApp components (services/queries with prebuilt UIs)” EX1005, ¶21.

177. An example hub is shown in Figure 18 of Rostoker, which illustrates a number of “xApps” (including “Data Services” components) displayed “on a first website” (e.g., <http://iltlvh161:56000/VC/default.jsp>) provided by content provider 101. EX1005, ¶67.



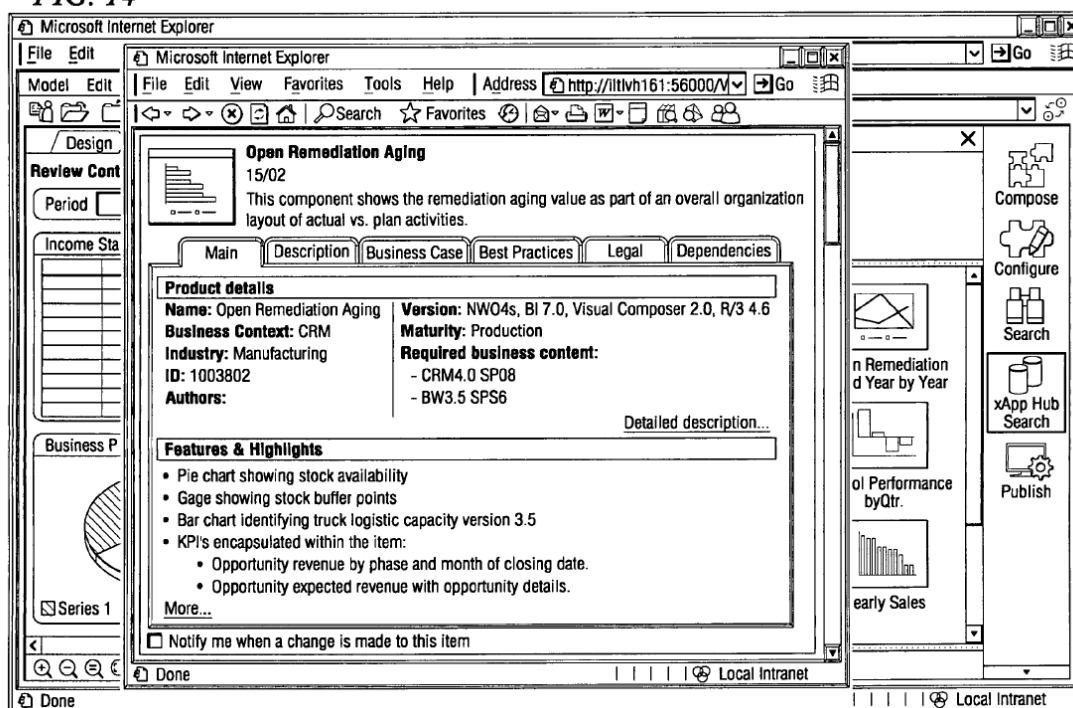
EX1005, FIG. 18 (annotated).

178. As illustrated in Figure 18, the xApps available on the “*first website*” include “*a plurality of . . . charts*,” for example the bar charts included in the “Open Remediation Aging” and “Control Performance by Qtr” xApps, and the line graphs included in the “Open Remediation Trend Year by Year” and “Period Balance” xApps. EX1005, FIG. 18. Figure 14 of Rostoker below describes that the “Open Remediation Aging” xApp includes a “[b]ar chart identifying truck logistic capacity.” EX1005, FIG. 14.



EX1005, FIG. 18 (annotated).

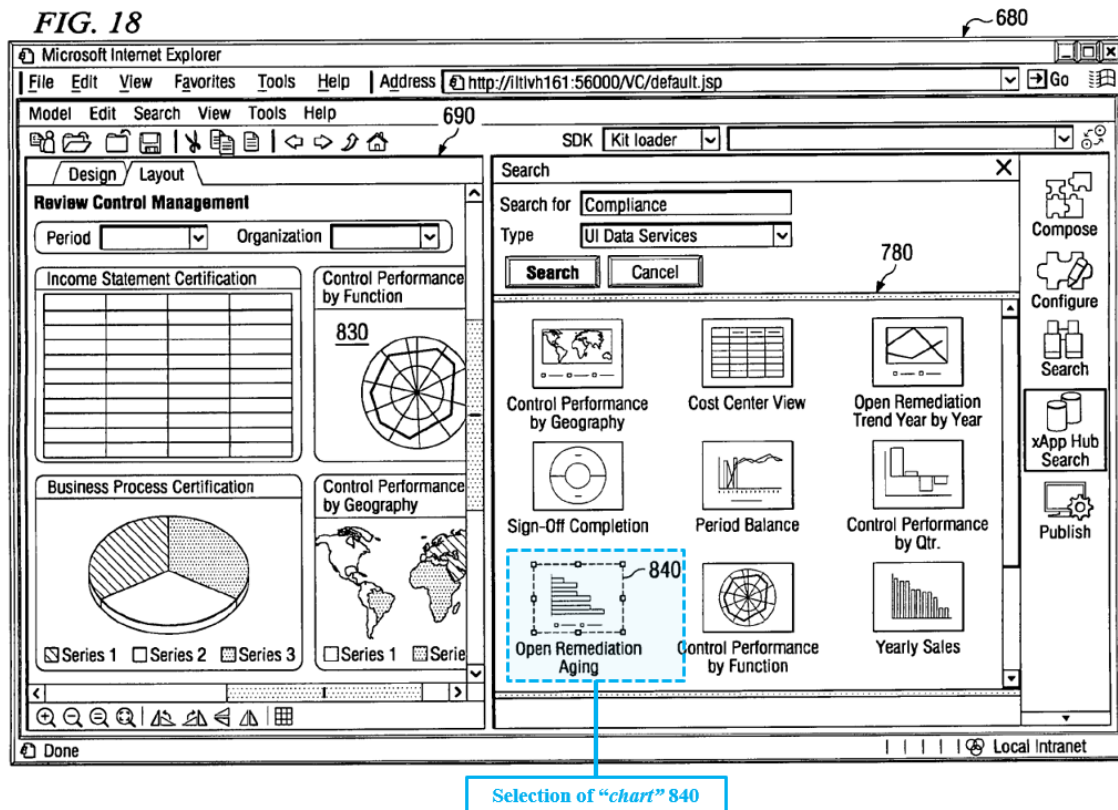
FIG. 14



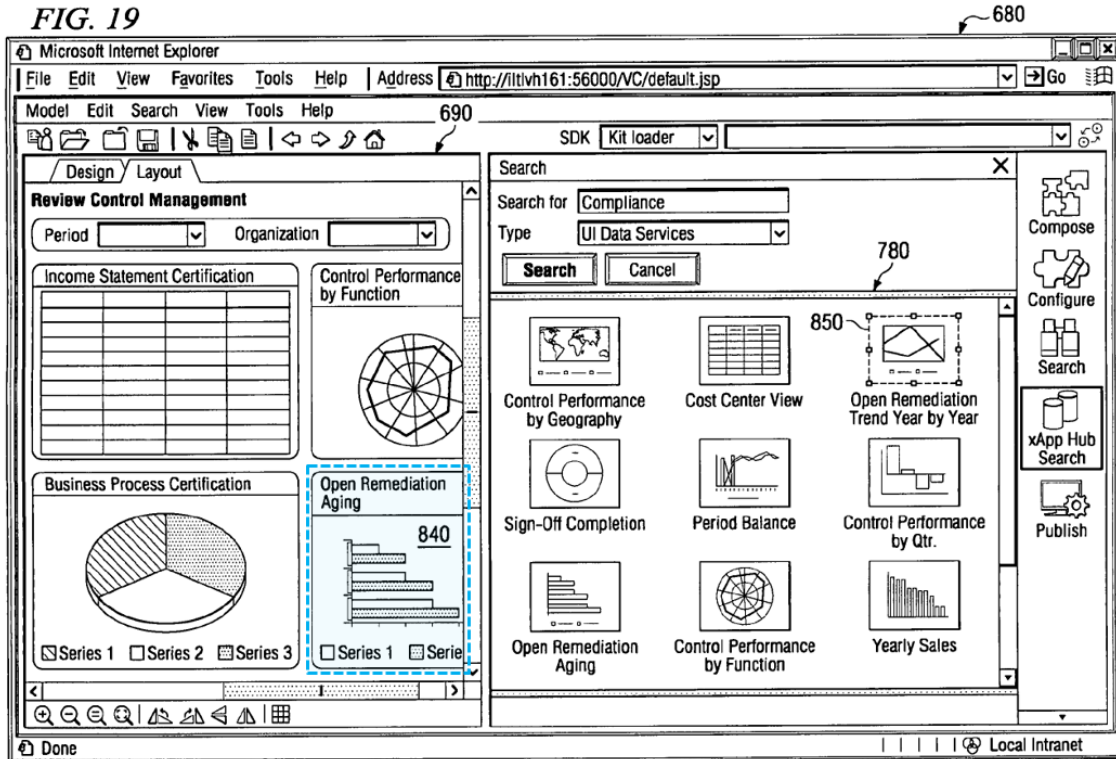
EX1005, FIG. 14.

179. Rostoker also discloses “receiving a selection, on a first website, of at least one . . . chart from a plurality of . . . charts.” As I discuss for element [1P], Rostoker discloses that a “user may locate [] components by searching the hub,” and “[t]he user may then select a pre-existing application or component and use the selected component or application without modification or use the component or application as a template, making modifications in order to create an application to operate in a desired manner.” EX1005, ¶6; see also EX1005, ¶8 (“A search may return one or more applications, application components, or other business process components that the user may import and configure to operate in a predetermined manner.”). Figures 18 and 19 of Rostoker (annotated and

reproduced below), for example, show selection of the “xApp 840” (“*at least one . . . chart from a plurality of . . . charts*”) by a user. EX1005, ¶67 (“xApp 840 [is] selected in the field 780 in FIG. 18. FIG. 19 shows the addition of the xApp 840 to the UI 690 as well as the selection of xApp 850 in the field 780.”), FIGs. 18, 19.



EX1005, FIG. 18 (annotated).

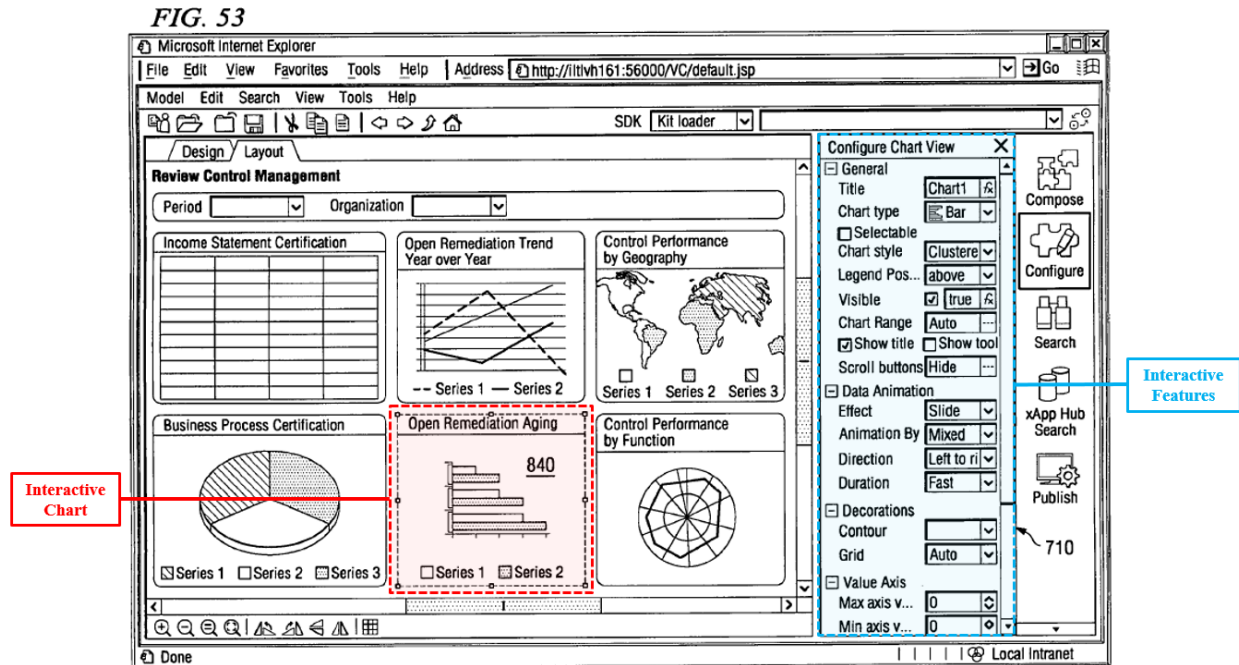


EX1005, FIGs. 18, 19 (annotated).

180. Therefore, it is my opinion that Rostoker discloses “*receiving a selection, on a first website, of at least one . . . chart from a plurality of . . . charts.*”

181. Rostoker further discloses that its charts are “*interactive charts*” in multiple ways. First, Rostoker discloses a “Configure Chart View” toolbar that allows a user to enable “*interactive*” features of the chart. For example, Figure 53 of Rostoker shows the “Configure Chart View” toolbar, which includes multiple “*interactive*” features, such as making the chart “Selectable” (suggesting interactivity using an input device such as a mouse), including “Scroll buttons” to scroll the chart, and adding “Data Animation[s].” EX1005, FIG. 53, ¶71. Figure 53

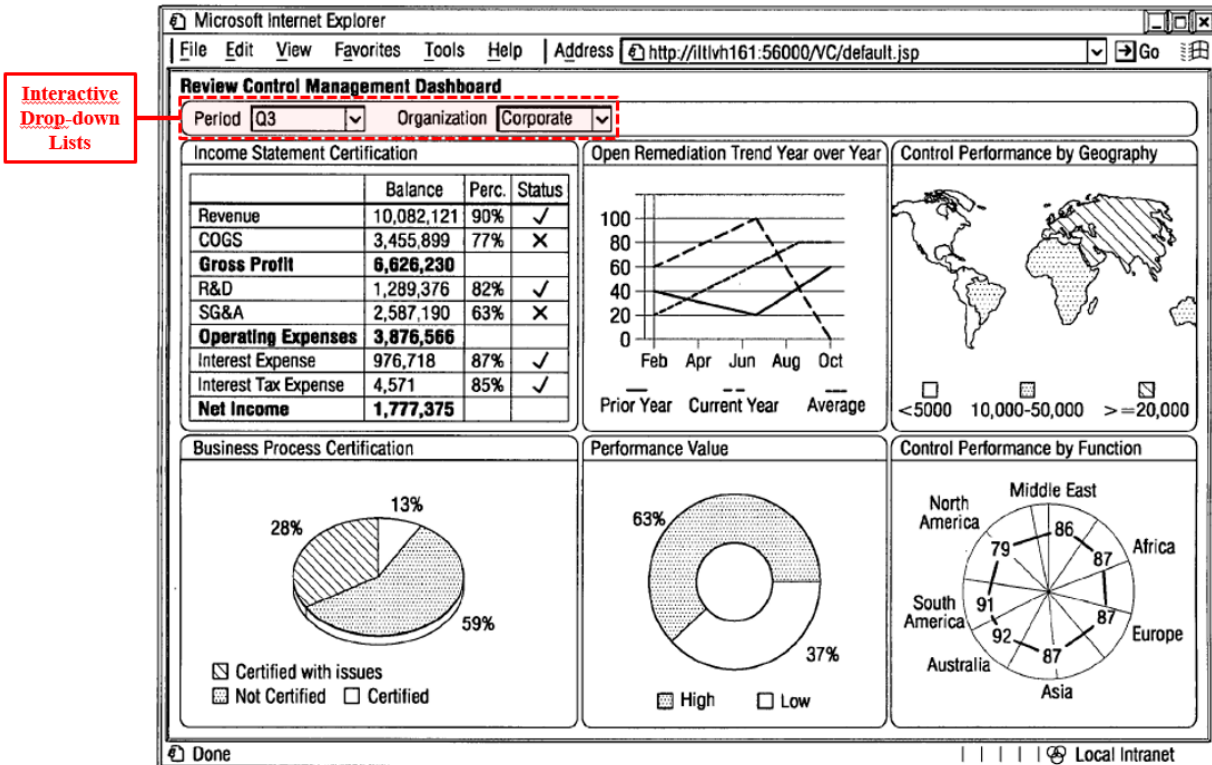
also shows that “xApp 840 is selected and the toolbar 710 lists the tools for configuring the xApp 840.” EX1005, ¶71.



EX1005, FIG. 53 (annotated).

182. Second, Figure 57 of Rostoker “shows the new or modified application after execution.” EX1005, ¶71. As illustrated in Figure 57, Rostoker’s executed application includes drop-down lists for “Period” (with a current value of “Q3”) and “Organization” (with a current value of “Corporate”). EX1005, FIG. 57. A POSA would also have understood that by changing the selection in a drop-down list (e.g., changing the “Period” from Q3 to Q2, and/or changing the “Organization” from “Corporate” to a different organization), the data in one or more of the selected charts (e.g., “Income Statement Certification” or “Performance Value,” as shown in Figure 57) would also change to reflect data of

a different period and/or a different organization. A POSA would have understood this type of interaction to be present within Rostoker's system based on its figures because it would have beneficially allowed users to view data specific to the selected "Period" and "Organization," as indicated by Rostoker's Figure 57—the drop-down lists would serve little purpose if they did not allow displayed charts to reflect the data selected in the drop down menus.

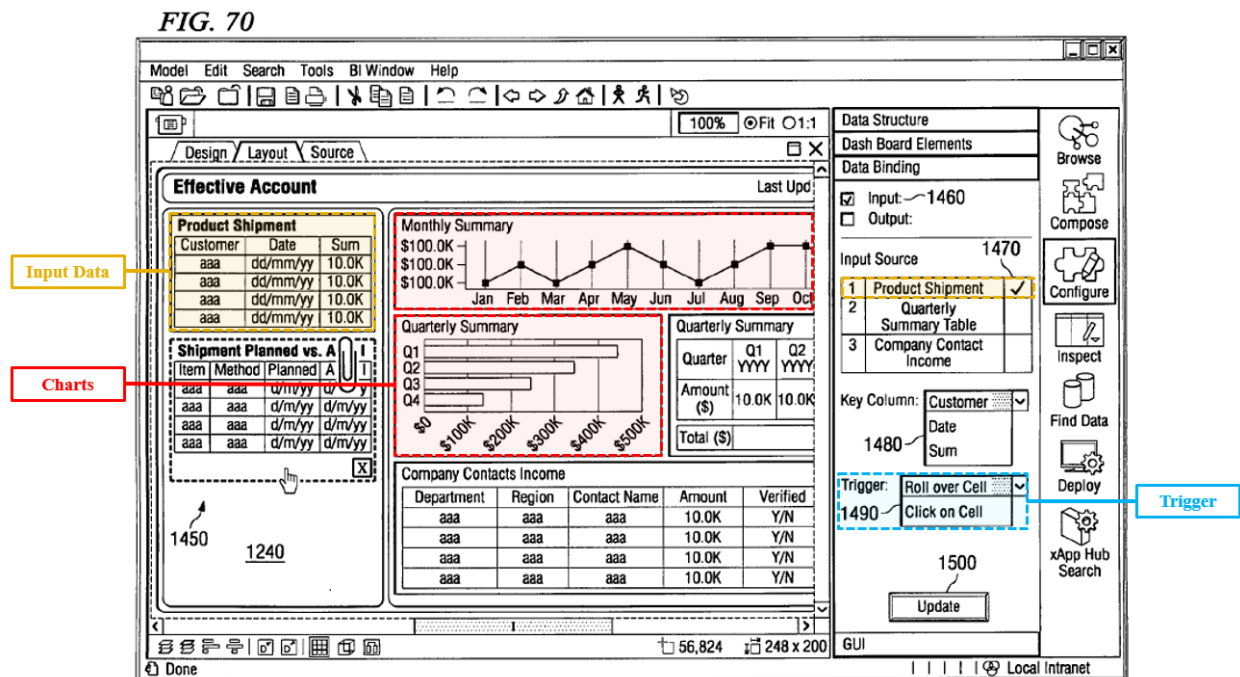
FIG. 57

EX1005, FIG. 57 (annotated).

183. It was well known that Flash enabled this type of functionality. For example, as Allaire describes, “[a]lmost all of the properties of Macromedia Flash’s intrinsic objects (textfields, graphics, movie clips) **can be dynamically**

modified at runtime using ActionScript.” EX1042, 0010. “This makes it easy to create highly customizable and adaptable interfaces.” EX1042, 0010.

184. Third, Rostoker discloses “trigger[s] to cause the input or output of [] data” to be associated with components. EX1005, ¶74, FIG. 70. For example, Figure 70 illustrates that Rostoker’s system allows a user to select “Input Source” (e.g., “Product Shipment data set”) to input to the application, select a “Key Column” (e.g., “Customer”) of the input data, and select a trigger (e.g., “Click on Cell”) to cause the input of the data. EX1005, ¶74, FIG. 70.

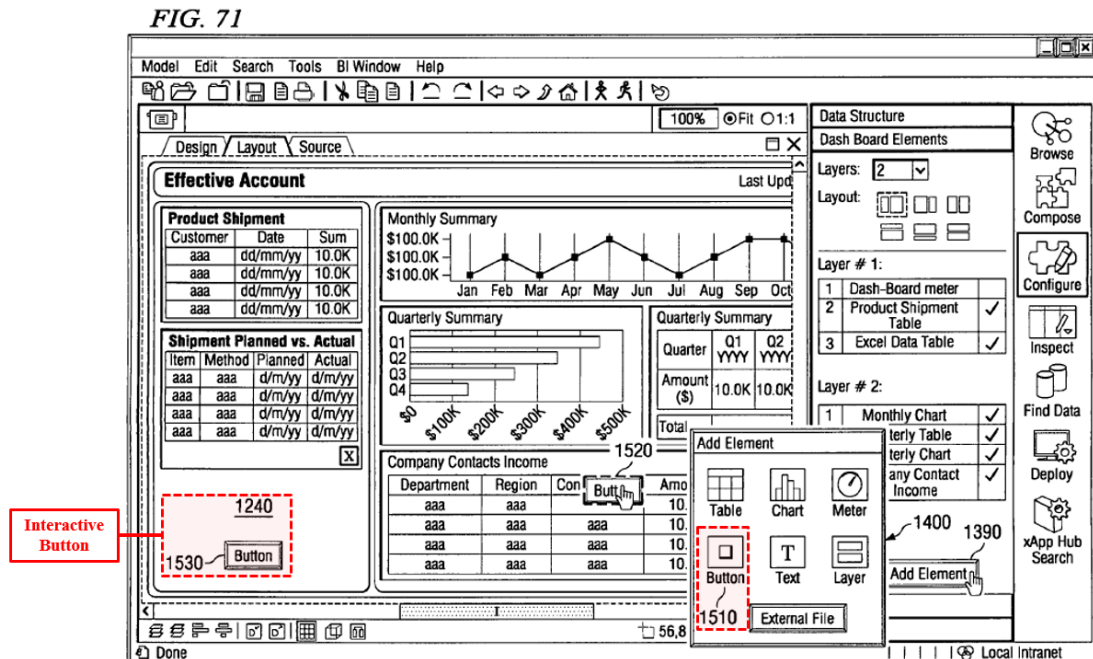


EX1005, FIG. 70 (annotated).

185. As I discuss above, in my opinion, a POSA would have understood that the “Click on Cell” trigger would allow a user to change the information displayed in the charts (e.g., “Monthly Summary” and “Quarterly Summary” of a

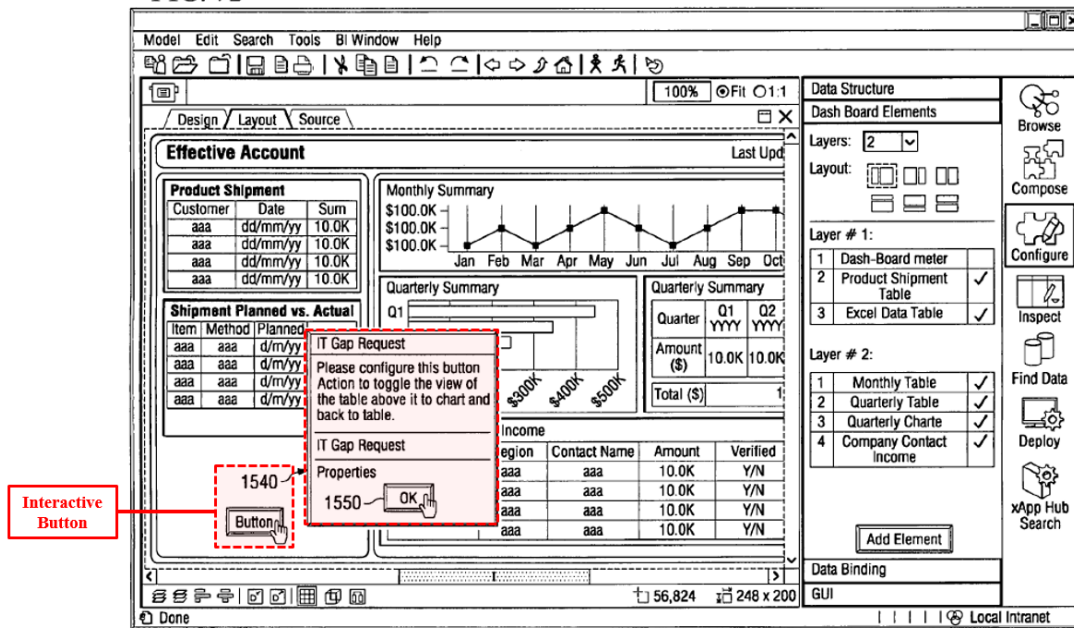
particular customer) when a user clicks a particular cell in the “Customer” column of the “Product Shipment” table. EX1005, FIG. 70. Such operation would be beneficial to allow the user to drill down into summary data for each individual customer, as depicted in Figure 70, annotated above.

186. Fourth, Rostoker discloses interactive buttons. For example, Figures 71 and 72 (annotated and reproduced below) of Rostoker illustrate that a user may configure “button object 1530” to, when clicked, “cause a table included with the application to switch between a table view and a chart view.” EX1005, ¶75, FIGs. 71, 72. In other words, the button allows a user to interactively show and hide the “chart view.” EX1005, ¶75, FIGs. 71, 72.



EX1005, FIG. 71 (annotated).

FIG. 72



EX1005, FIGs. 71, 72 (annotated).

187. It is my opinion that the types of interactive features taught by Rostoker are consistent with those disclosed in the '000 patent. As I discuss above, the '000 patent describes “interactive features that allow a user to interact with the chart to, for example, modify or change their view of the chart, learn more about the chart data, and the like.” EX1001, 9:9-13. And the '000 patent discloses that these “interactive features” can include, for example, “a zoom feature, a data mouse-over pop-up window, a data hide/share feature, an audio playback feature, a drill down feature, a data gliding feature and the like.” EX1001, 9:13-16. As I discuss above, Rostoker’s interactive options provide similar “data hide/share” and “drill down features,” as well as additional “interactive features” within the scope of the claim. EX1001, 9:9-16.

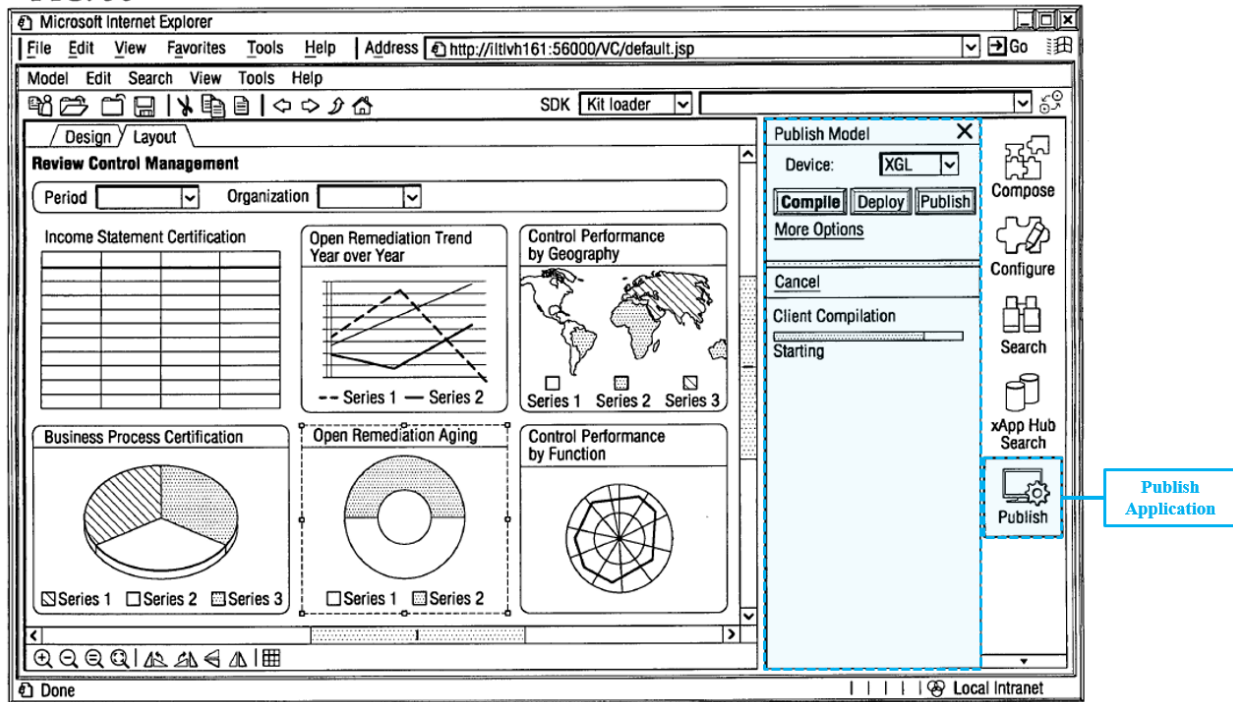
188. Therefore, it is my opinion that Rostoker discloses “*receiving a selection, on a first website, of at least one interactive chart from a plurality of interactive charts.*”

3. [1B] “displaying, in response to the selection the at least one interactive chart embedded in a chart box on a second website, wherein the chart box is a web widget”

189. Element [1B] recites “*displaying, in response to the selection the at least one interactive chart embedded in a chart box on a second website, wherein the chart box is a web widget.*” In my opinion, Rostoker discloses this limitation.

190. As I discuss for element [1A], Rostoker teaches “*receiving a selection, on a first website, of at least one interactive chart.*” See Section IX.A.2; EX1005, ¶67, FIGs. 18, 19. Once charts are selected and added to the current application, Rostoker discloses a “Publish” function that allows the user to publish the “new or modified application,” as illustrated in Figure 56. EX1005, ¶71, FIG. 56.

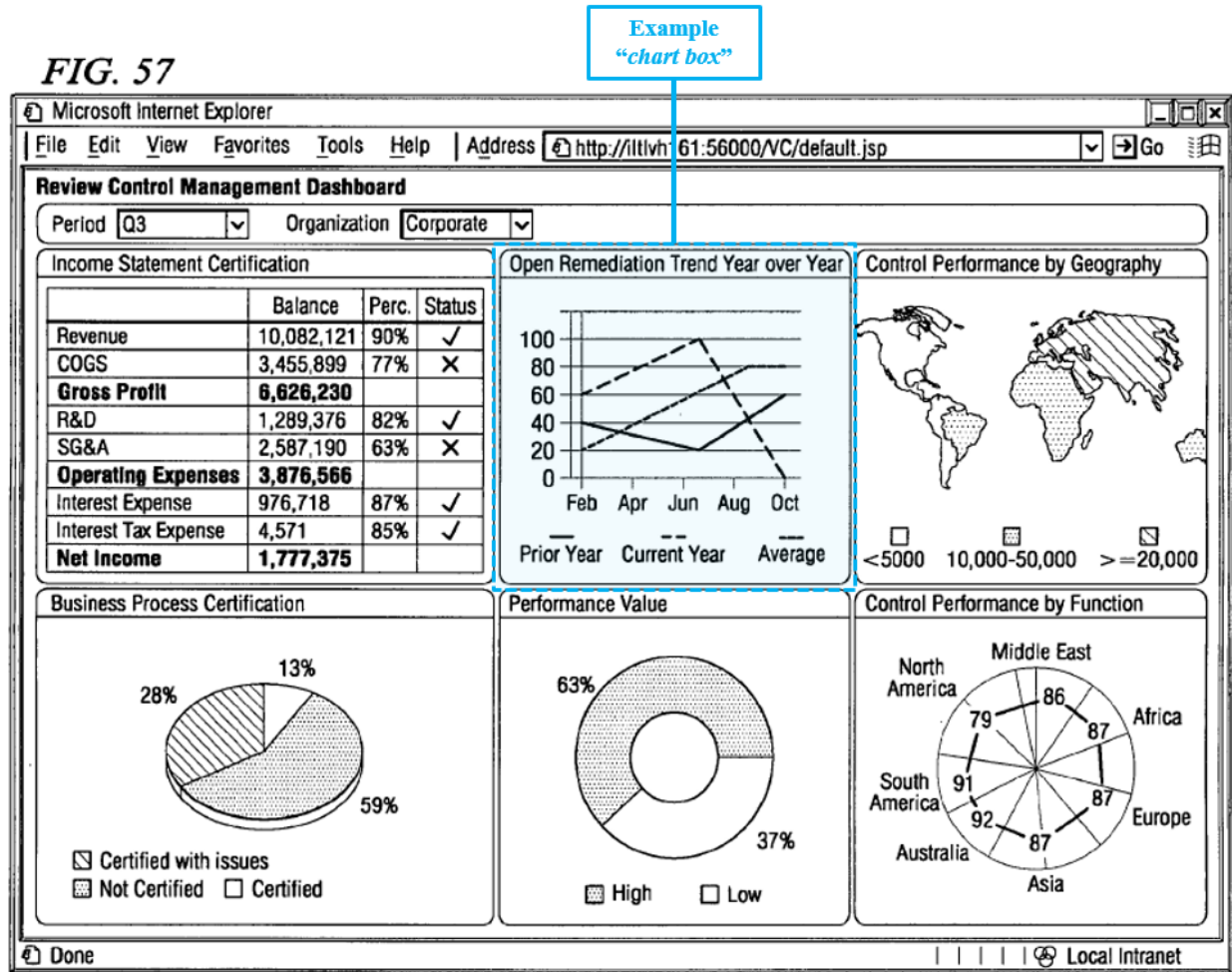
FIG. 56



EX1005, FIG. 56 (annotated).

191. Rostoker explains, “the Publish button . . . caus[es] each of the xApps included in the new application to be executed.” EX1005, ¶71. “FIG. 57 shows the new or modified application after execution.” EX1005, ¶71.

192. As illustrated in Figure 57 below, each selected “*interactive chart*” is displayed in a “*chart box*” in the executed application, for example, the box labeled “Open Remediation Trend Year over Year.” EX1005, FIG. 57.

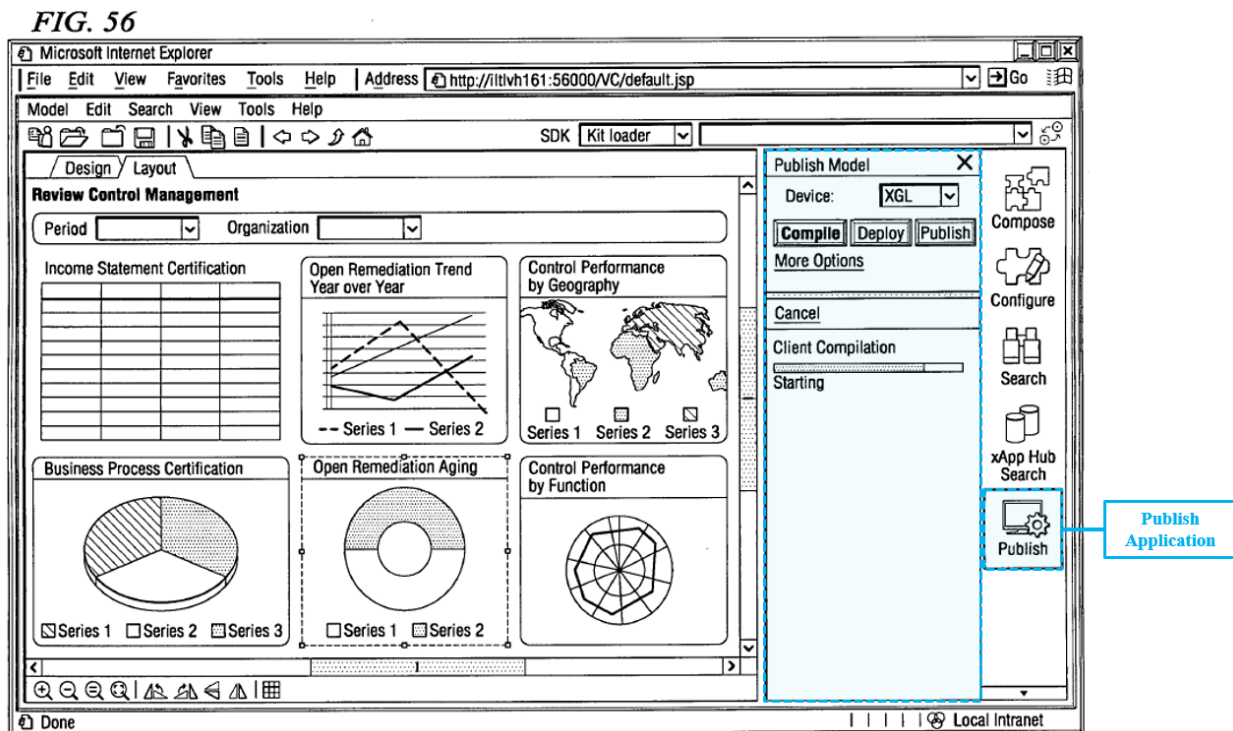


EX1005, FIG. 57 (annotated).

193. I understand Rostoker's chart box is the same as the chart box described in the '000 patent. The '000 patent discloses that the chart box is "configured to be embedded on a website to display one or more interactive charts on that website," EX1001, 15:52-55, which is the only guidance provided in the '000 specification. Similar to that description, Rostoker's chart box is configured to be embedded on a website, as shown in Figure 57 above, to display one or more interactive charts on that website. Thus, it is my opinion that Rostoker discloses

“displaying, in response to the selection the at least one interactive chart embedded in a chart box.” EX1005, FIG. 57.

194. It is my opinion that Rostoker also discloses “the chart box is a web widget.” For example, Rostoker discloses that each “chart box” in the application illustrated in Figure 57 is its own “xApp.” EX1005, ¶71 (describing “each of the xApps included in the new application”). As shown in Figure 56, as part of publishing the application and included xApps, the application components are “transformed to XGL [(‘executable GUI language’)] representations,” which are then “used to generate a machine-executable runtime GUI” to be executed by a target device. EX1005, ¶¶42, 46.



EX1005, FIG. 56 (annotated).

195. Rostoker explains that “[t]he XGL representation may be used for generating representations of various different GUIs and supports various GUI features including full windowing and componentization support, rich data visualizations and animations, rich modes of data entry and user interactions, and flexible connectivity to any complex application data services.” EX1005, ¶43. Rostoker further discloses “an XGL-to-Java compiler” to “generate Java code [] for execution by a target device comprising a Java runtime,” and “an XGL-to-Flash compiler” to “generate Flash code [] for execution by a target device comprising a Flash runtime.” EX1005, ¶50.

196. The ’000 patent describes that “[t]he chart box may be a flash widget” without defining what a widget is. EX1001, 2:45, 16:1. It is my understanding that Rostoker describes a “*widget*” that is consistent with the description in the ’000 patent. The World Wide Web Consortium (W3C) defines a “*widget*” as “an end-user’s conceptualization of an interactive single purpose application for displaying and/or updating local data or data on the Web, packaged in a way to allow a single download and installation on a user's machine or mobile device.” EX1065, 0002. “A widget may run as a stand alone application (meaning it can run outside of a Web browser), or may be embedded into a Web document.” EX1065, 0002. Thus, a POSA would have understood that Rostoker’s published xApps, which are Java

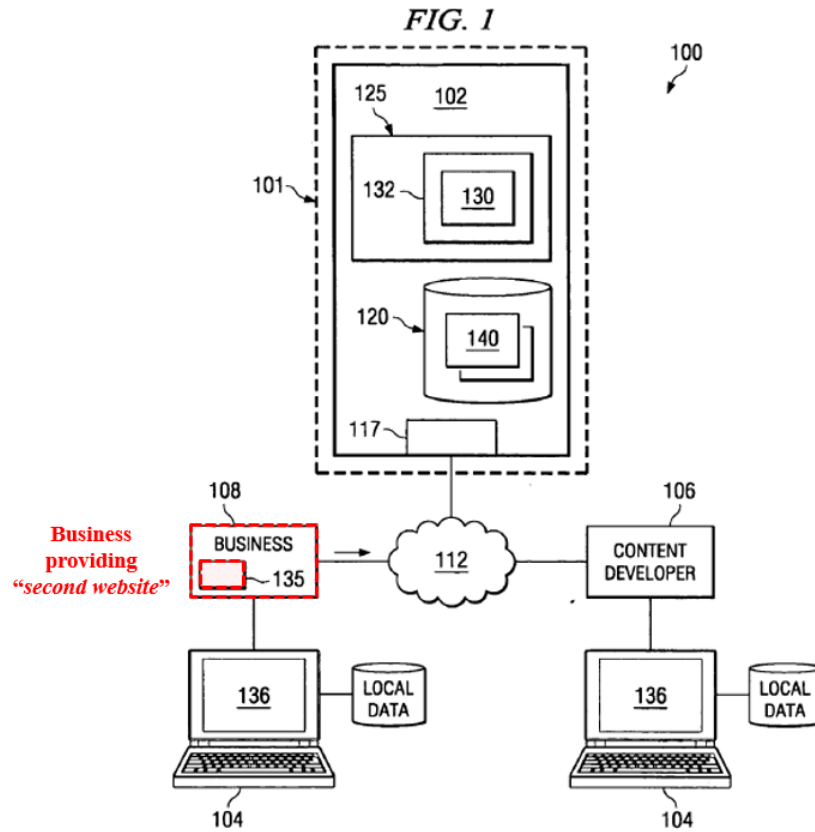
or Flash codes for execution and display within a website, as illustrated, for example, in Rostoker's Figure 57, are "*widget[s]*." EX1005, ¶50, FIGs. 2A, 57.

197. Additionally, it is my opinion that Rostoker discloses "*displaying, in response to the selection the at least one interactive chart embedded in a chart box on a second website.*" For example, as I have described for element [1A], Rostoker discloses that a user may select "*at least one interactive chart*" to create an application using "*a first website.*" See Section IX.A.2; see also EX1005, ¶¶5-6 ("[T]he user is capable of creating, adapting, and configuring an analytical application or dashboard to create a new business application" by "select[ing] a pre-existing application or component and us[ing] the selected component or application without modification or . . . making modifications in order to create an application to operate in a desired manner"); see also EX1005 ¶40 ("The modeling environment 132 may be utilized to create, adapt, or configure the data services 140 (available to a user via the component manager 130) for creating a custom application, such as a business application, to fulfill a need of the user.").

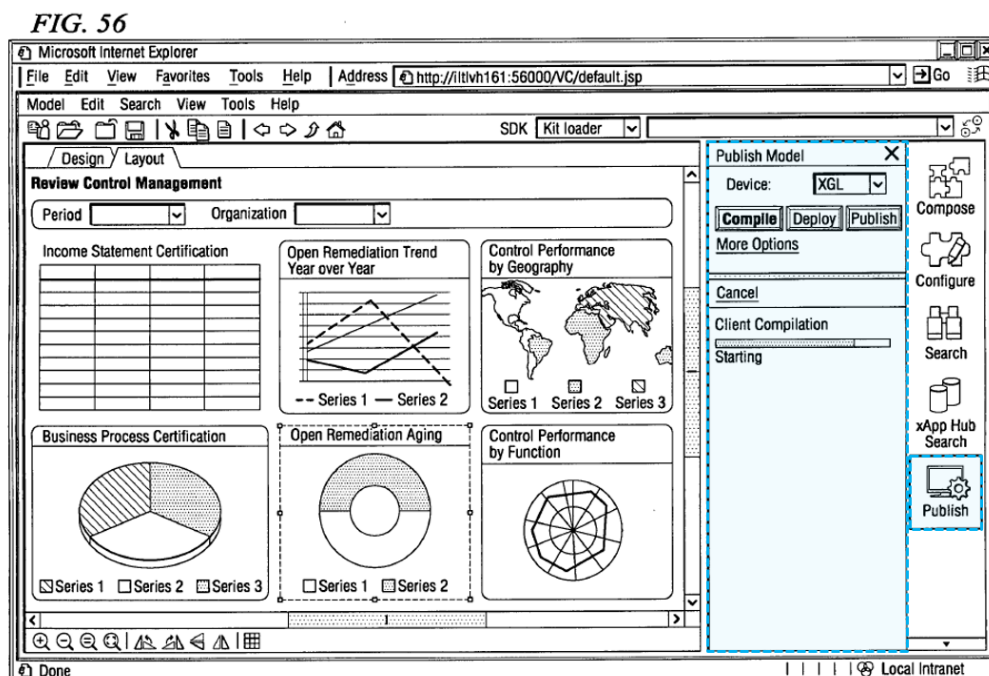
198. Once an application is created, Rostoker teaches deploying and storing the application within a business environment. Figure 1 of Rostoker below, for example, illustrates a business 108, separate from content provider 101 (which provides the claimed "*first website*"), storing "composite applications 135 (or

xApps).” EX1005, ¶¶21-22, FIG. 1. And Figure 57 illustrates Rostoker’s

“Publish”/“Deploy” functions. EX1005, FIG. 56.



EX1005, FIG. 1 (annotated).

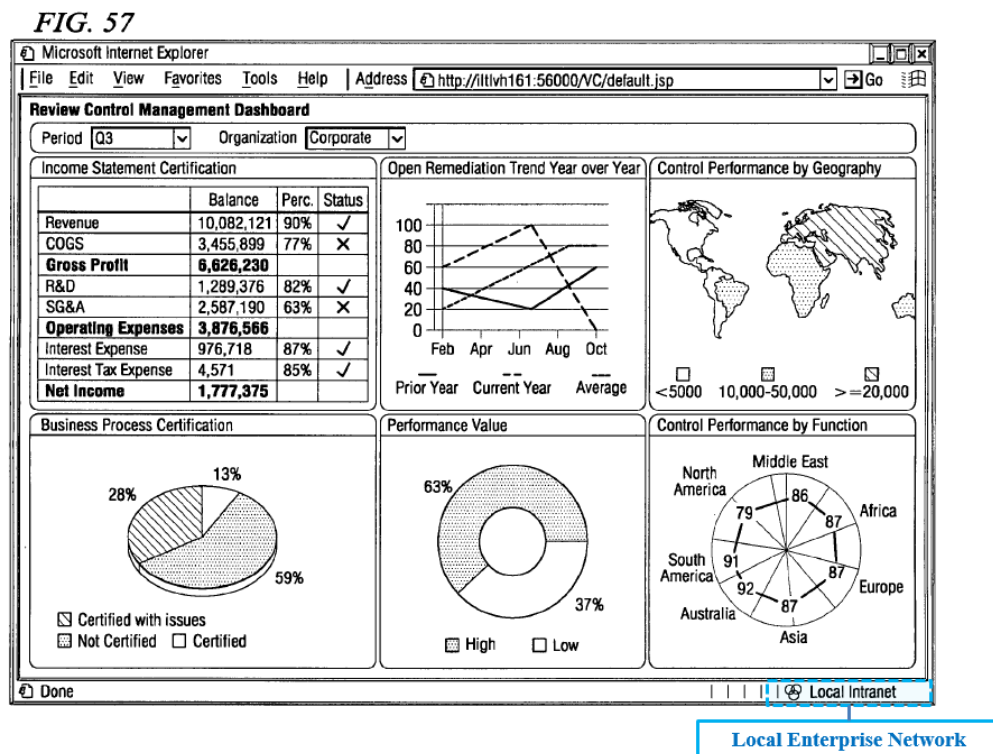


EX1005, FIG. 56 (annotated).

199. Rostoker discloses that its system “present[s] a plurality of data services 140 to businesses 108, or their individual users, as well as other companies and users in the network.” EX1005, ¶19. As I discuss above, Rostoker’s system allows its users (e.g., “businesses 108, or their individual users”) to create a new business application by using or modifying “pre-existing application[s] or component[s].” EX1005, ¶6. Consistent with Rostoker’s architecture, at the time of the ’000 patent, a POSA would have understood that businesses, such as business 108 of Rostoker, often provide their own web services to make business applications accessible within the business’s local network. For example, Nikolaidou describes that “Enterprise information systems (EISs) consist of a combination of interrelated **Intranet-based** and Internet-based applications, built

on multi-tiered client-server models.” EX1064, 0001. “The Web platform,” which includes a “web server,” “serves well as **a common access environment for all the applications operating within enterprise Intranet boundaries.**” EX1064, 0001.

200. Indeed, Rostoker’s “Publish” function allows the user to “Deploy” an application, such as deploying “applications 135” to an enterprise network associated with the business. EX1005, FIG. 56. And, consistent with Rostoker’s “Deploy” functionality, Rostoker’s Figure 57 indicates that the published application is accessible via a “Local Intranet,” which in the case of a “business application” may be a local enterprise network. EX1005, ¶¶4, 28.



EX1005, FIG. 57 (annotated).

201. A POSA would have been motivated to allow business applications to be deployed to a separate website local to an enterprise to ensure published business applications are not accessible outside of the local enterprise network. *See, e.g.*, EX1064, 0001 (discussing “applications operating within enterprise Intranet boundaries”; explaining that “[s]ecurity issues should also be considered.”). And, in such a case, a POSA would have understood the published application (for example, as shown in Rostoker’s Figure 57 above) to be accessed and displayed on “*a second website*” provided by the enterprise network, rather than by Rostoker’s content provider 101.

202. Moreover, providing Rostoker’s published applications “*on a second website*” would have simply been a design choice resulting in the same function of displaying the published application within a web browser. Both Intranet-based and Internet-based applications were well known and used by enterprises at the time of the ’000 patent, as evidenced, for example, in Nikolaidou. EX1064, 0001. In deciding on an application architecture for a particular enterprise, “[s]ecurity issues should also be considered,” and providing enterprise applications through a local web server (within an enterprise Intranet) would have acted to increase security of who may access those applications. EX1064, 0001. Therefore, providing published applications through “*a second website*” within an enterprise

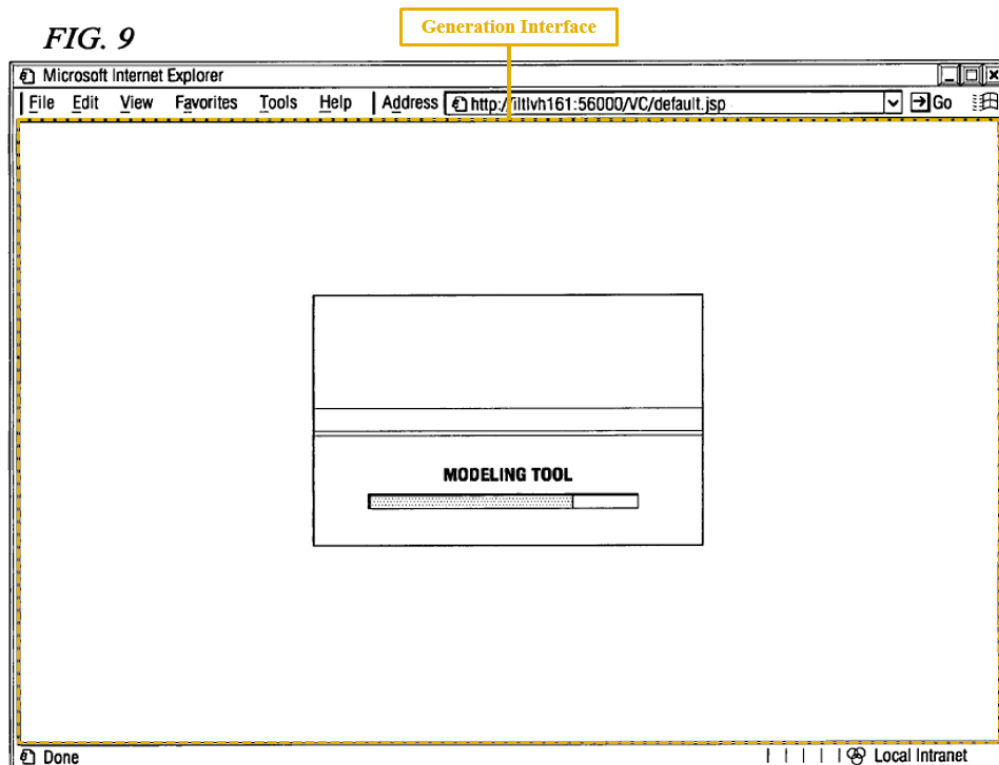
would have, at a minimum, been an obvious choice depending on the needs of the organization.

203. Therefore, it is my opinion that Rostoker teaches “*displaying, in response to the selection the at least one interactive chart embedded in a chart box on a second website.*”

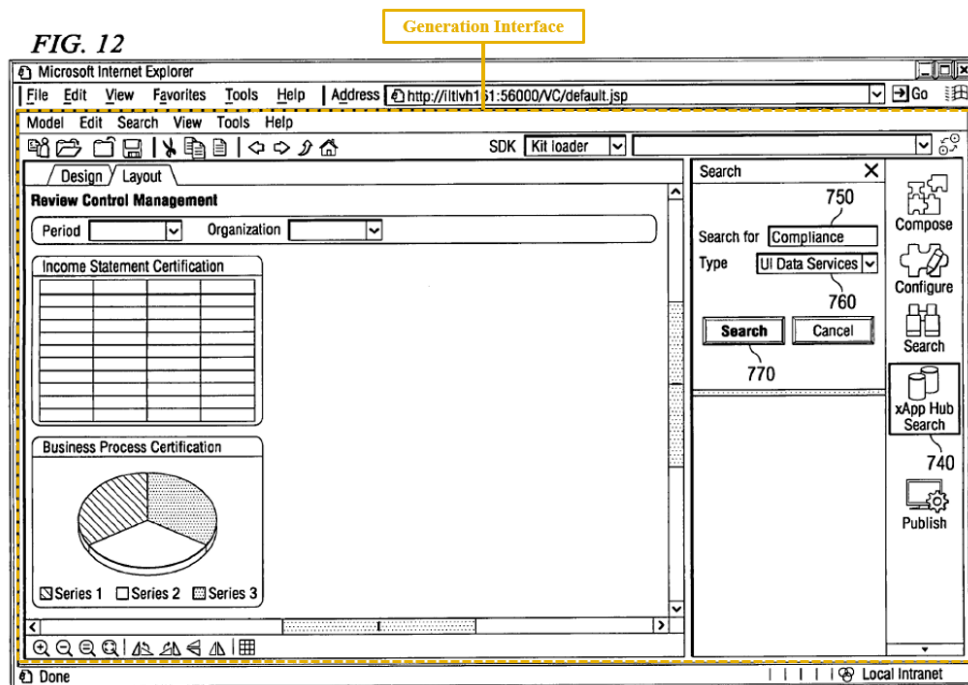
4. [1C] “**wherein the at least one interactive chart is generated on a generation interface configured to allow a user to generate the interactive chart with a chart template from a chart template store**”

204. Element [1C] recites “*wherein the at least one interactive chart is generated on a generation interface configured to allow a user to generate the interactive chart with a chart template from a chart template store.*” In my opinion, Rostoker discloses this limitation.

205. It is my opinion that Rostoker teaches a “*generation interface*” in the form of a “**modeling environment 132.**” EX1005, ¶40. The “modeling environment” is shown throughout Rostoker’s figures, for example in Figures 9 and 12.



EX1005, FIG. 9 (annotated).



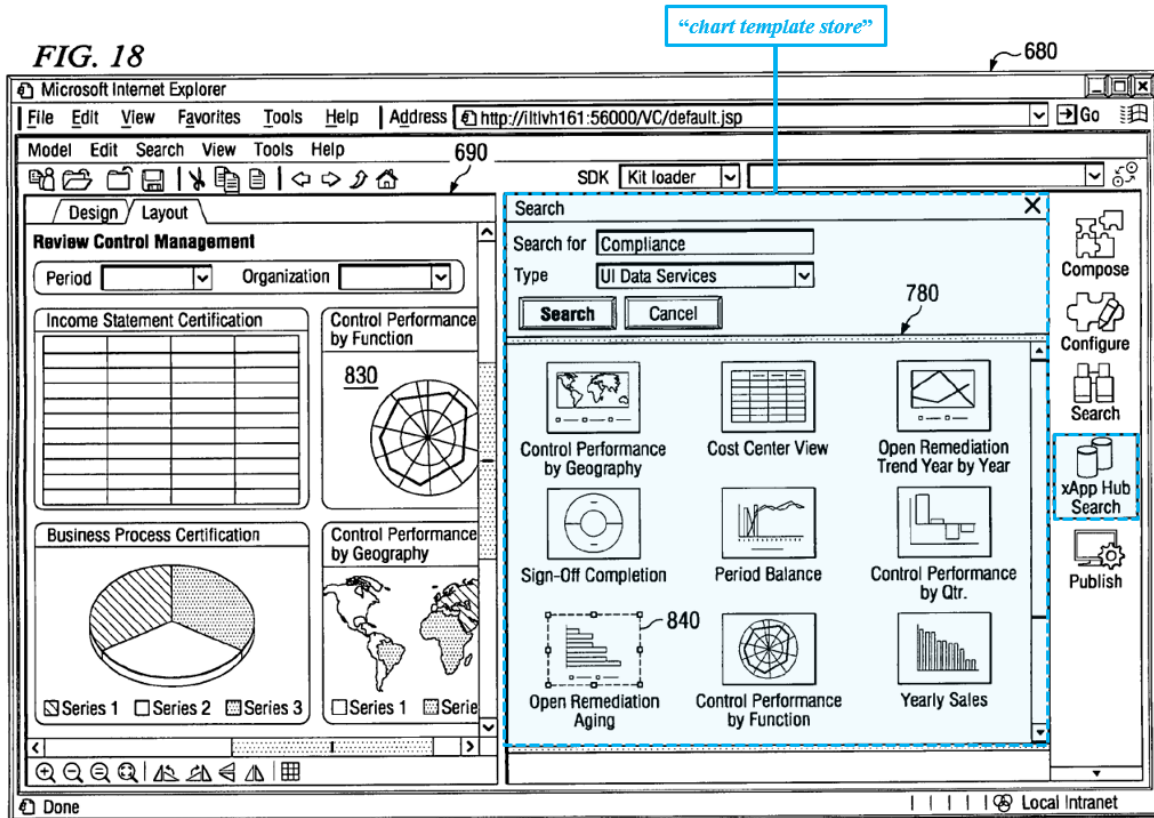
EX1005, FIG. 12 (annotated).

206. Rostoker discloses that “[t]he modeling environment 132 may be utilized to create, adapt, or configure the data services 140 (available to a user via the component manager 130) for creating a custom application, such as a business application, to fulfill a need of the user,” and “[t]he applications may include applications created by the developers with the modeling environment 132.”

EX1005, ¶40. Rostoker further discloses that created components and applications can be made available via Rostoker’s hub/storefront, as I discuss for element [1A]. For example, Rostoker discloses that a “user may locate the components by searching the hub,” and “[t]he user may then select a pre-existing application or component” EX1005, ¶6. Thus, it is my opinion that Rostoker teaches that “*the at least one interactive chart is generated on a generation interface,*” specifically, the modeling environment 132.

207. Rostoker also teaches that the “*generation interface [is] configured to allow a user to generate the interactive chart with a chart template from a chart template store*” in at least two ways. First, Rostoker discloses “a **storefront** having a knowledge base or compilation of software components, applications, or other data services that can easily be modified or integrated into different business analyst and end user tools.” EX1005, ¶6. From Rostoker’s “storefront”/“hub” a “user may [] select a pre-existing application or component and . . . use the

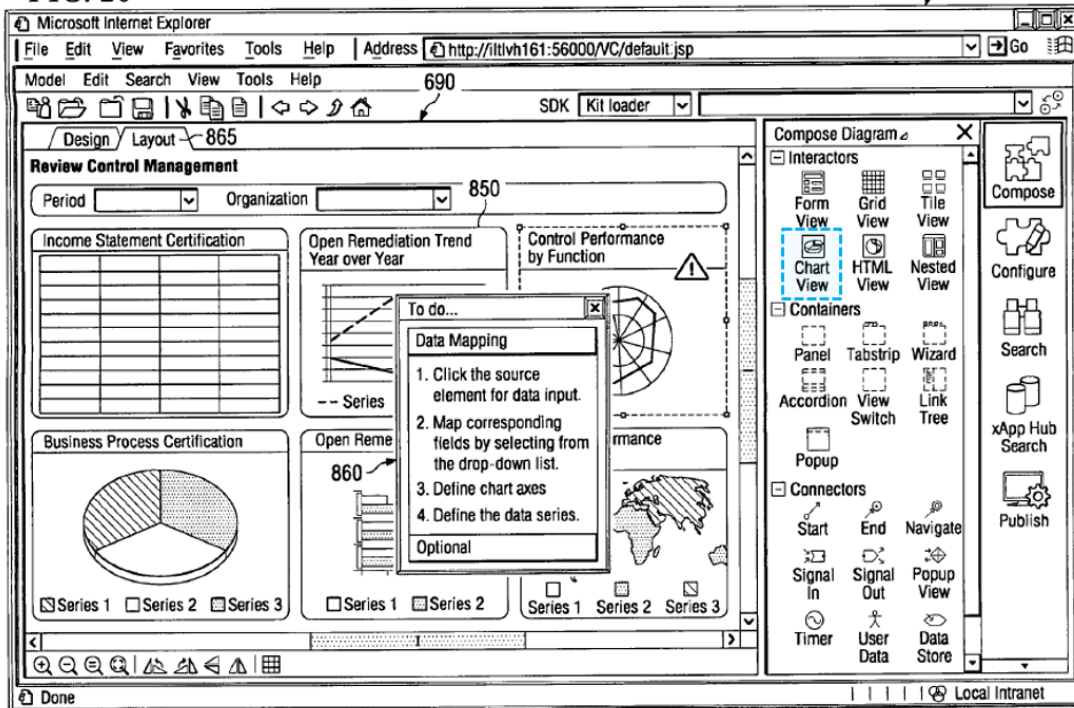
component or application as a template, making modifications in order to create an application to operate in a desired manner.” EX1005, ¶6. An example of Rostoker’s “storefront”/“hub” (i.e., “*chart template store*”) is shown in Figure 18 below.



EX1005, FIG. 18 (annotated).

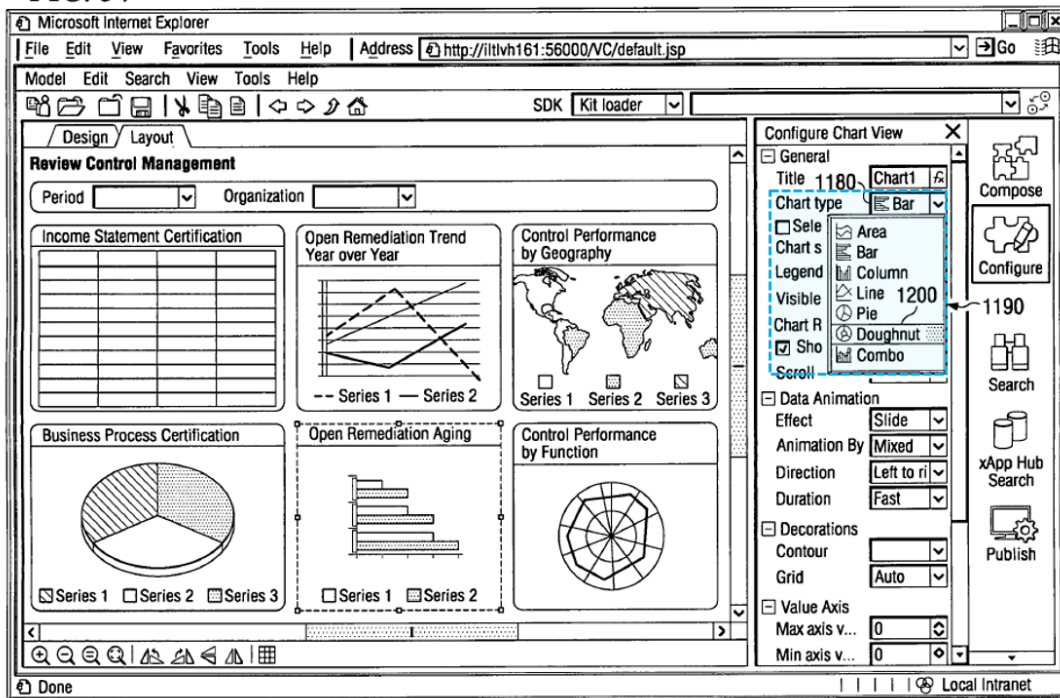
208. Second, Rostoker teaches selecting a new chart view, using a “[c]hart type” as a template, as illustrated in Figures 20 and 54 below.

FIG. 20



EX1005, FIG. 20 (annotated).

FIG. 54



EX1005, FIG. 54 (annotated).

209. As Rostoker explains:

[T]he chart type drop-down list 1180 is shown and **numerous available graphical representations of chart types 1190 are shown**. Accordingly, the user is able to change the chart type of the selected xApp 840 by merely selecting the desired type 1190 from the list 1180.

EX1005, ¶71.

210. Rostoker also discloses that “data services 140” and “any other appropriate data” are stored in memory and data service 140 “may include . . . composite applications, analytics, **reusable components**, [and] objects”

EX1005, ¶¶30, 31. A POSA would have understood that data pertaining to these “chart types” (i.e., “*chart template[s]*”) are stored in memory (i.e., a “*chart template store*”) either at a “central location” or within “local data at the user’s site,” as described in Rostoker. EX1005, ¶¶28, 71. In other words, Rostoker’s reusable templates must be stored in memory so that they are accessible to the user developing an application. Thus, in my opinion, Rostoker also teaches that its “*generation interface [is] configured to allow a user to generate the interactive chart with a chart template from a chart template store*” using a selected “chart type.” EX1005, ¶¶28, 71.

211. Therefore, it is my opinion that Rostoker discloses this claim limitation.

5. [1D] “wherein the at least one interactive chart, when displayed on the second website in the absence of the generation interface, enables a user to a) render the chart, b) obtain input from a user and c) rerender the chart using the input thereby enabling the interactive chart to retain interactivity”

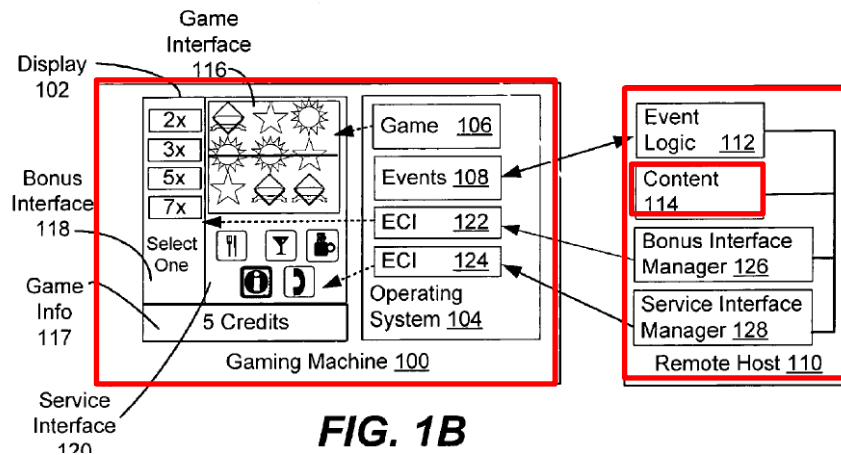
212. Element [1D] recites “*wherein the at least one interactive chart, when displayed on the second website in the absence of the generation interface, enables a user to a) render the chart, b) obtain input from a user and c) rerender the chart using the input thereby enabling the interactive chart to retain interactivity.*” In my opinion, Rostoker teaches this limitation.

213. As I have explained for element [1A], Rostoker teaches that its charts contain “*interactive*” features, such as changing the selection in a drop-down list and clicking buttons that “cause a table included with the application to switch between a table view and a chart view.” See Section IX.A.2; EX1005, ¶¶75, FIGs. 53, 57, 70-72. These interactive features “*b) obtain input from a user and c) rerender the chart using the input*” because the chart is modified in response to the user input, as I have explained for element [1A]. See Section IX.A.2.

214. Additionally, as I discuss for element [1B], Rostoker teaches “*enabl[ing] a user to a) render the chart*” such that it is “*displayed on the second website in the absence of the generation interface.*” See Section IX.A.3. In particular, as I have explained for element [1B], Rostoker teaches publishing applications and included xApps as Java or Flash objects “for execution by a target

device comprising” either “a Java runtime” or “a Flash runtime.” *See* Section IX.A.3; EX1005, ¶¶42-43, 50.

215. It is well known that flash files are self-contained and can be executed independent of any generation interface. For example, LeMay discloses a remote host that downloads multimedia content including flash files. EX1051, ¶55. The flash files may be transmitted to and displayed at a second device, such as gaming machine 100, depicted below. EX1051, ¶54.



EX1051, FIG. 1B (annotated).

216. Similarly, Peters discloses downloading flash movies onto an electronic reading device from a server. EX1052, ¶¶25, 68, 72. Peters also discloses that the user is able to view the Flash file (i.e., the movie) when the electronic reading device loses connection to the server. EX1052, ¶72.

217. Therefore, A POSA would have understood that Rostoker’s published applications, including “*interactive chart[s]*” would be displayed “*on the second*

website” using the Java or Flash runtime, independent of the interface on the first website used to create the application.

218. Accordingly, it is my opinion that Rostoker discloses “*wherein the at least one interactive chart, when displayed on the second website in the absence of the generation interface, enables a user to a) render the chart, b) obtain input from a user and c) rerender the chart using the input thereby enabling the interactive chart to retain interactivity.*”

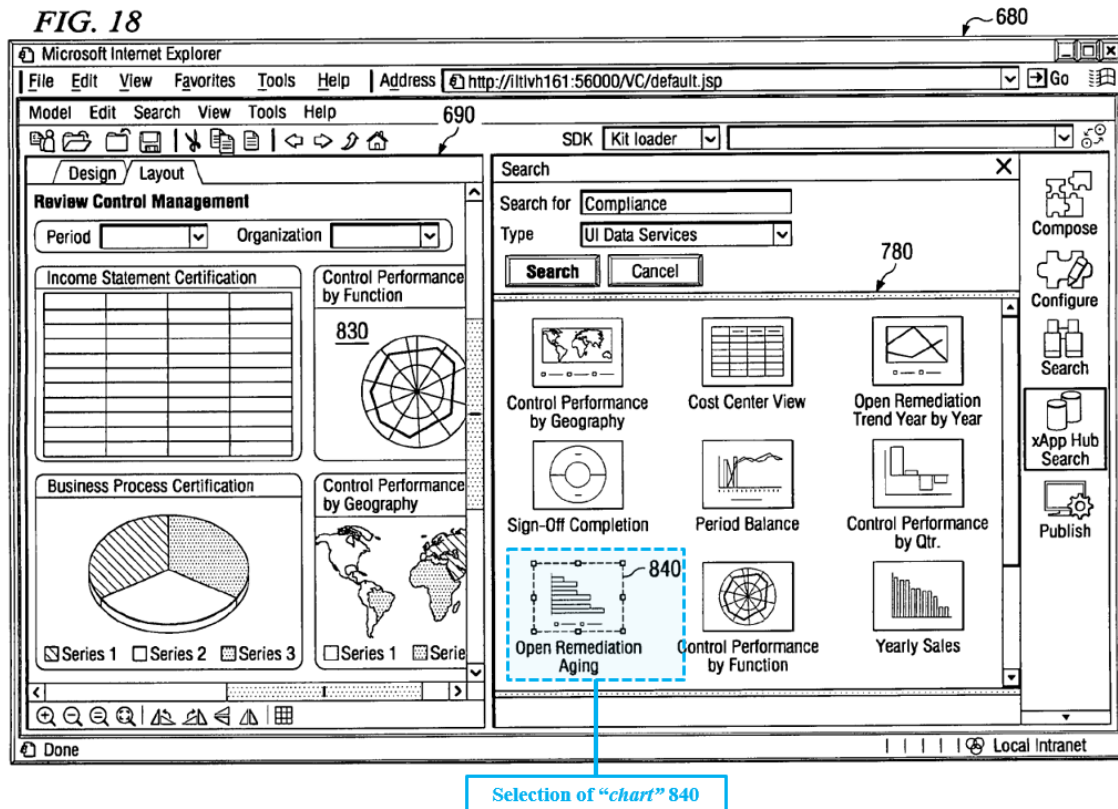
6. [1E] **“wherein the selection is made by dragging and dropping the at least one interactive chart from the plurality of charts into a chart box folder connected to the chart box such that retrieve code for data feed displays the at least one interactive chart on the first website”**

219. Element [1E] recites “*wherein the selection is made by dragging and dropping the at least one interactive chart from the plurality of charts into a chart box folder connected to the chart box such that retrieve code for data feed displays the at least one interactive chart on the first website.*” In my opinion, Rostoker teaches this limitation.

220. It is my opinion that Rostoker discloses “*wherein the selection is made by dragging and dropping the at least one interactive chart from the plurality of charts.*” As I discuss for element [1A], a user may locate an “*interactive chart*” using Rostoker’s hub, and “[t]he user may then **select a pre-**

existing application or component,” as illustrated in Figure 18 below. EX1005,

¶6.

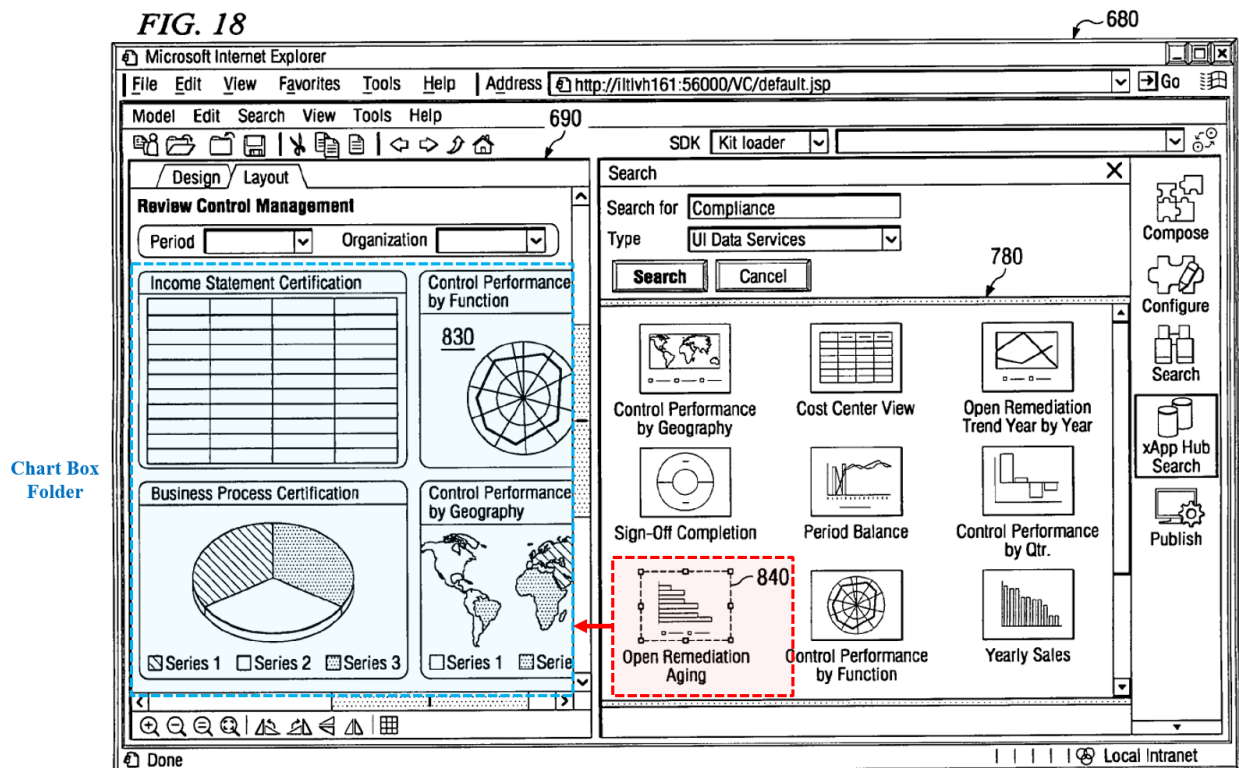


EX1005, FIG. 18 (annotated).

221. Rostoker further discloses that a user may “conduct[] a search of the hub contents and select[] a component corresponding to the need of the new application.” EX1005, ¶61. “The new component, object, or feature may be dragged and dropped or otherwise imported into the application.” EX1005, ¶6; see also EX1005, ¶59 (“[T]he user may utilize the analytics workbench to modify or otherwise configure the selected xApp by, for example, dragging and dropping desired features or pre-constructed objects from a toolbar onto the UI of the

xApp.”). Thus, Rostoker teaches that the “*the selection is made by dragging and dropping.*”

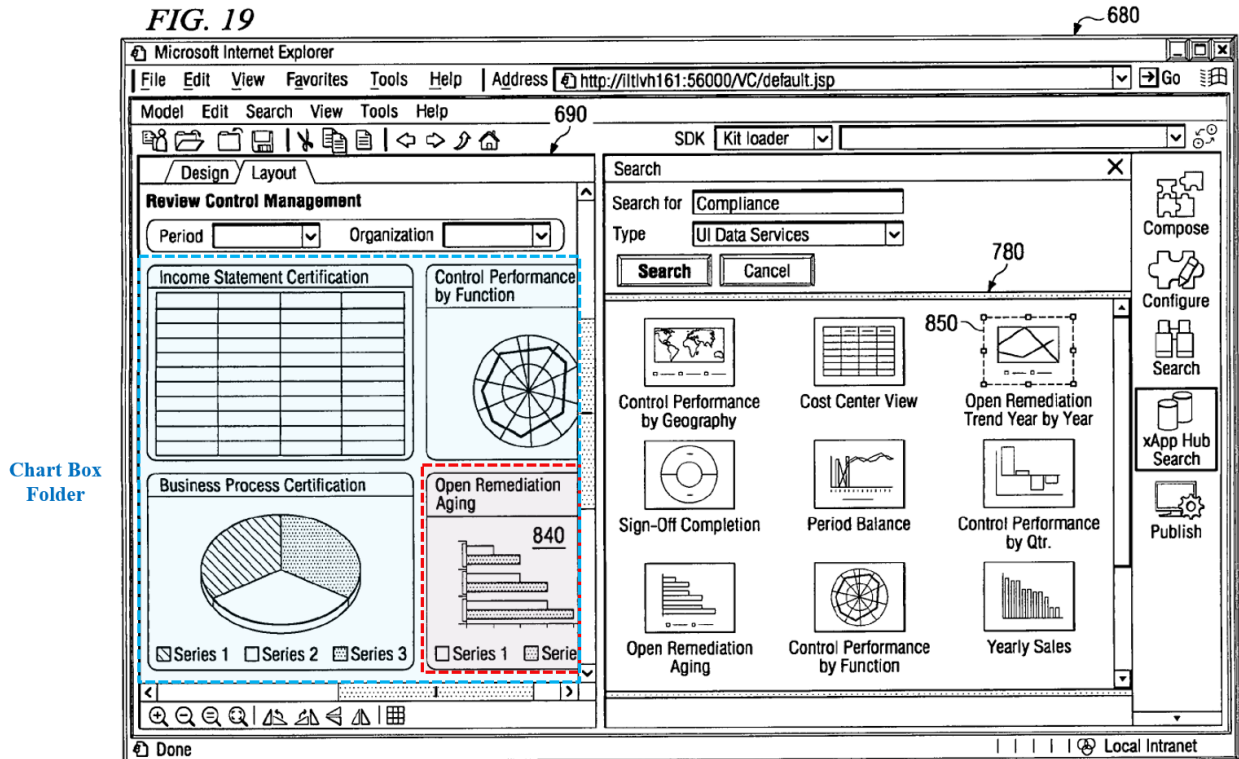
222. It is also my opinion that Rostoker discloses that “*the selection is made by dragging and dropping the at least one interactive chart from the plurality of charts into a chart box folder connected to the chart box.*” For example, the Layout tab of an application in Rostoker’s modeling environment provides a “*chart box folder*” associated with the application, as illustrated in Figure 18 below.



EX1005, FIG. 18 (annotated).

223. As shown in Figure 18 above, the “*chart box folder*” contains “*interactive chart[s]*” that are part of the current application, such as, “Income

Statement Certification,” “Business Process Certification,” etc. Figure 19 of Rostoker below illustrates that when chart 840 (“Open Remediation Aging”) is dragged and dropped into Rostoker’s “*chart box folder*,” the chart is added to the folder.

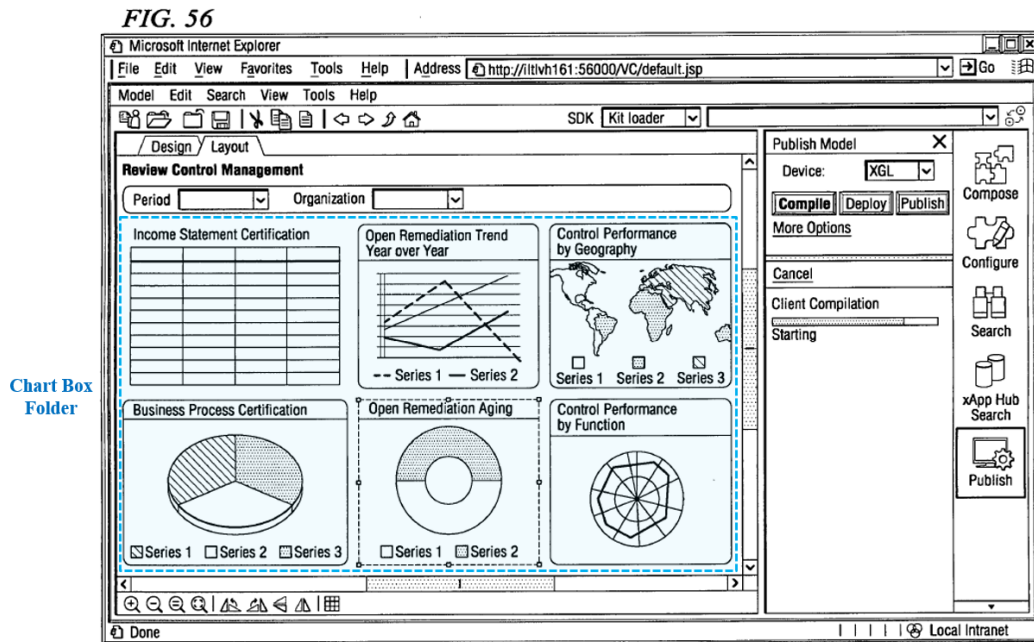


EX1005, FIG. 19 (annotated).

224. A POSA would have also understood that the “*chart box folder [is]* connected to the chart box” because, when published and executed, the application will display the charts contained in the “*chart box folder*” within “*chart box[es]*” in the executed application. For instance, as illustrated in Figures 56 and 57 below, each chart placed into the “*chart box folder*” for an application within Rostoker’s modeling environment (as depicted in Figure 56) is displayed within a “*chart box*”

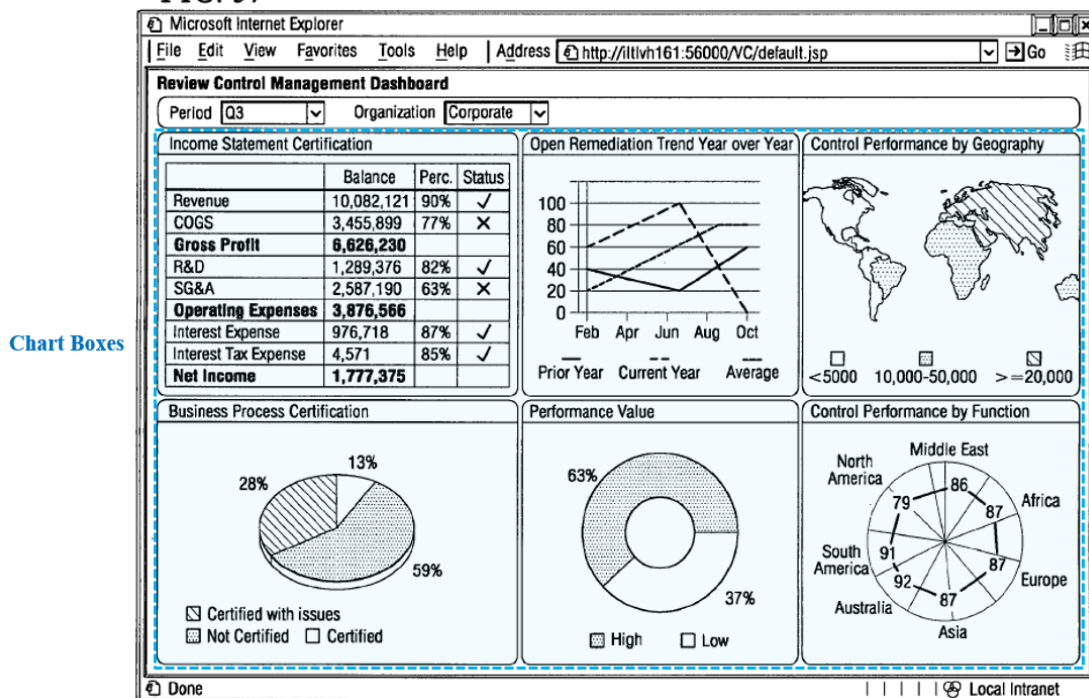
of the published and executed application (as depicted in Figure 57). EX1005, ¶71

(“In FIG. 56, the Publish button 1210 is selected, causing each of the xApps included in the new application to be executed or otherwise prototyped. FIG. 57 shows the new or modified application after execution.”).



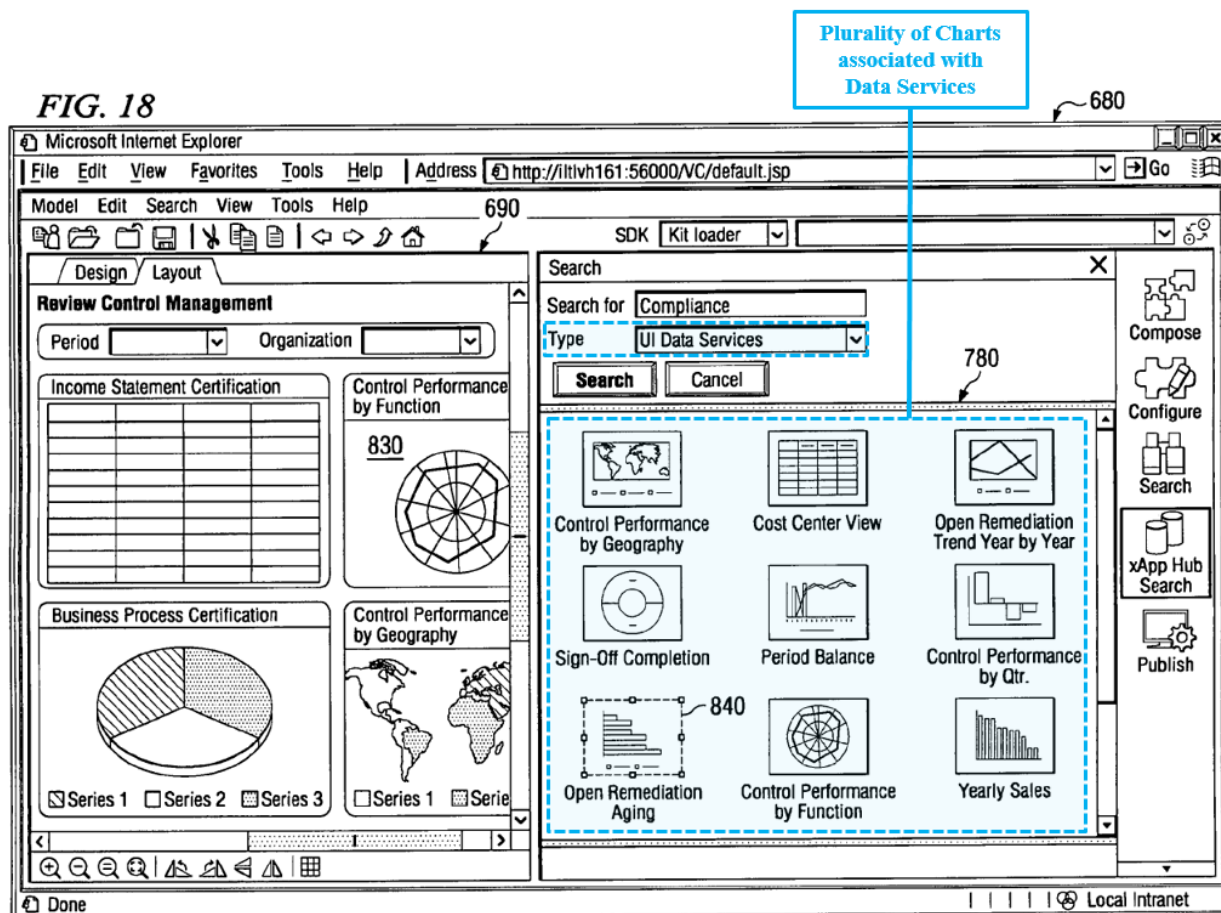
EX1005, FIG. 56 (annotated) (showing interface for creating application).

FIG. 57



EX1005, FIG. 57 (annotated) (showing execution of published application).

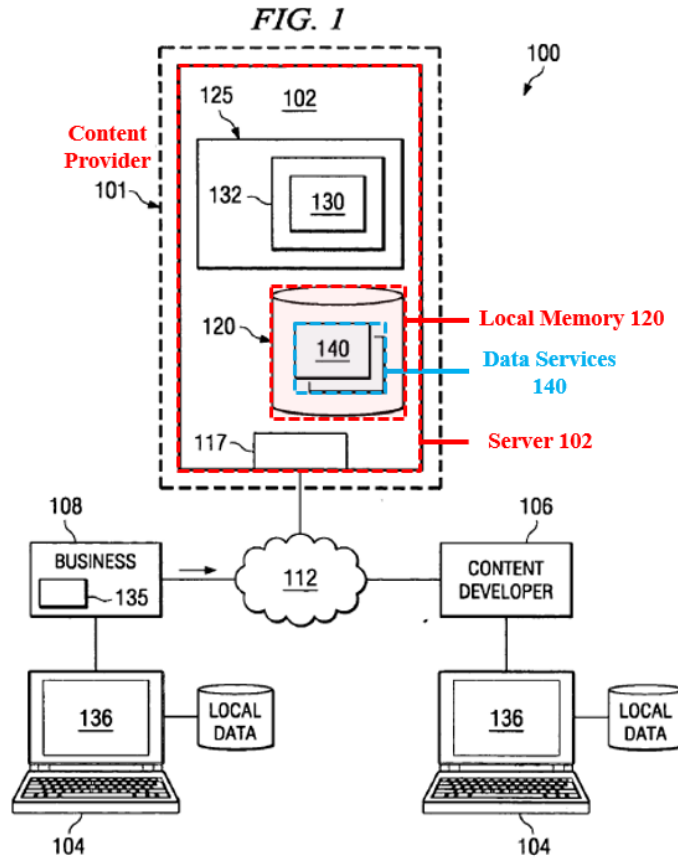
225. Additionally, it is my opinion that Rostoker teaches “*such that retrieve code for data feed displays the at least one interactive chart on the first website.*” Rostoker discloses that its charts can be associated with “data services”: “Generally, the content provider 101 uses a component manager 130, or other similar information hub of enterprise knowledge and components, to **present a plurality of data services 140** to businesses 108, or their individual users, as well as other companies and users in the network.” EX1005, ¶19. “[D]ata services 140 may include . . . composite applications, analytics, reusable components, objects, [and] . . . dashboards” EX1005, ¶31. Figure 18 of Rostoker below illustrates a plurality of charts associated with data services 140.



EX1005, FIG. 18 (annotated).

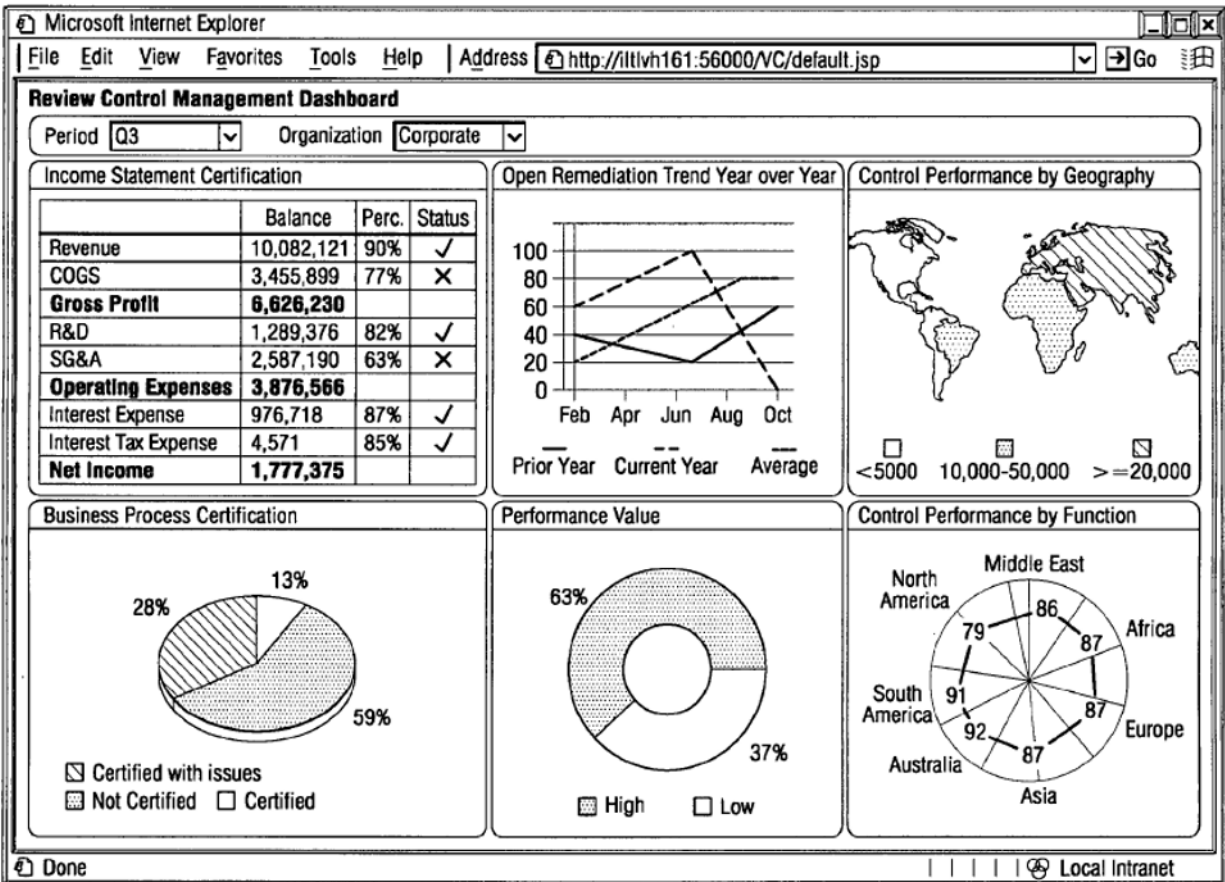
226. Rostoker discloses that “[d]ata services 140 (and their accompanying information) may be stored in one or more logical or physical repositories.”

EX1005, ¶32. And “data services 140 may comprise one table or file or a plurality of tables or files stored on one computer or across a plurality of computers in any appropriate format.” EX1005, ¶32. Figure 1 of Rostoker below illustrates that the data services may be stored within “local memory 120” of server 102. EX1005, ¶30.



227. Figure 57 of Rostoker below illustrates that the “*interactive chart[s]*” included in the application retrieve and display data from the associated data service. Indeed, Rostoker explains that its “[r]eal-time dashboards often provide table and graph information on the current state of the data, which may be supplemented by data services 140.” EX1005, ¶39.

FIG. 57



EX1005, FIG. 57.

228. A POSA would have, therefore, understood that Rostoker’s “interactive chart[s]” contain “retrieve code for [a] data feed,” i.e., code for retrieving real-time data from the data service, which is then used to “display[] the at least one interactive chart,” as illustrated above in the displayed charts of Figure 57. And this “retrieve code” is executed when the user executes the application, for example when “the Publish button [] is selected.” EX1005, ¶71. Therefore, it is my opinion that Rostoker teaches that “retrieve code for data feed displays the at least one interactive chart on the first website.”

229. Therefore, it is my opinion that Rostoker teaches “*wherein the selection is made by dragging and dropping the at least one interactive chart from the plurality of charts into a chart box folder connected to the chart box such that retrieve code for data feed displays the at least one interactive chart on the first website.*”

7. [1F] “**wherein the at least one interactive chart is automatically displayed on the second website using a data feed from the first website to the second website upon the dragging and dropping selection.**”

230. Element [1F] recites “*wherein the at least one interactive chart is automatically displayed on the second website using a data feed from the first website to the second website upon the dragging and dropping selection.*” In my opinion, the combination of Rostoker and Strong teaches this limitation.

231. As I discuss for element [1E], Rostoker teaches “*wherein the selection is made by dragging and dropping the at least one interactive chart from the plurality of charts into a chart box folder.*” See Section IX.A.6. As I discuss for element [1B], “*upon the dragging and dropping selection,*” the application may be published and deployed to “*the second website,*” e.g., a website provided by a business’s enterprise network. See Section IX.A.3. Rostoker does not expressly disclose that “*the at least one interactive chart is **automatically displayed** on the second website . . . **upon the dragging and dropping selection.***” However, it is my opinion that a POSA would have understood that automatically publishing

applications upon modification (e.g., selecting a new chart) would reduce the burden on developers to publish updated applications, and ensure that users of Rostoker's application are always viewing the latest version, as I discuss in further detail below. These motivations would have led a POSA to a reference like Strong, which discloses automatically distributing data objects when a modification to the data object is made.

232. Specifically, Strong discloses techniques for “synchronizing a data object amongst the clients.” EX1067, ¶26. Strong discloses that when a change/update is made to a data object, the data object is automatically published to sharing clients: “[C]hanges or updates 30 from a sharing client 34 will be sent to the synchronization engine 32 which will forward or update a copy of a data object on each of the clients listed as part of the plurality of sharing clients 34 in order to synchronize that data object with other sharing clients.” EX1067, ¶26. For example, as Strong explains, **“if changes are made by one of the sharing clients 34 to a particular data object, an update message 30 is sent from the particular sharing client to the synchronization engine 32.”** EX1067, ¶27. Then, **“[t]he synchronization engine, in turn, will make the particular adjustment to the original or copy of the data object on the server and distribute the data object out to each of the sharing clients 34.”** EX1067, ¶27.

233. In other words, Strong discloses that when modifications are made to a data object, the modifications will automatically be synchronized with the server, which in turn publishes the data object to other clients “so that the most recent versions are available to the users.” EX1067, ¶¶26-28, 31. This process obviates the need for the user to explicitly publish the updated data object to other sharing clients—all clients viewing the data object will always be viewing the latest version, rather than an outdated version.

234. Although Strong describes its techniques primarily in the context of “[f]ile sharing,” EX1067, ¶3, a POSA would have recognized that Strong’s techniques would apply to distribution of any data object, including Rostoker’s applications, which can take the form of Java or Flash objects. EX1005, ¶50. Applying Strong’s techniques to Rostoker’s application publishing process would enable applications to be automatically published and deployed upon detection of a change, such as dragging and dropping a chart into the application (i.e., “*the at least one interactive chart is automatically displayed on the second website . . . upon the dragging and dropping selection*”). As I discuss above for elements [1A] and [1E], when an “*interactive chart*” is selected for inclusion in an application, for example by “dragging and dropping . . . pre-constructed objects from a toolbar onto the UI of the xApp,” Rostoker’s system detects the selection and modifies the application to include the new chart. EX1005, ¶¶59, 61, FIGs. 18, 19 (showing

selection and addition of chart “Open Remediation Aging” to an application); *see* Sections IX.A.2, IX.A.6.

235. Following modification of the application, Strong’s techniques would enable the application to be automatically published and deployed. As I noted above, a POSA would have been motivated to do so for at least two reasons. First, Strong’s automatic distribution techniques would reduce the burden on developers to publish updated applications, obviating the need for developers to go through the “Publish” toolbar depicted in Rostoker’s Figure 56. Instead, this process could automatically initiate and run in the background as the developer continues work on the application, increasing development efficiency. Second, as Strong explains, incorporation of Strong’s techniques would ensure that “that the most recent versions are available to the users” of Rostoker’s applications. EX1067, ¶31. Such a process would allow users to view chart updates (and their associated business insights) in real-time (or near real-time).

236. At a minimum, publishing applications upon detection of a change would have been an obvious variant of Rostoker’s operation, automatically publishing an application rather than waiting for a user to expressly click the publish button. The core functionality would remain the same in each variant—after modification to an application, the application is published, and users of the

application may view the updated application. The only difference is the timing of when the application is published.

237. The combination, therefore, would have been nothing more than using a known technique (distribution of a data object upon modification) to improve a similar device (Rostoker's application development and publishing system). A POSA also would have had a reasonable expectation of success incorporating Strong's automatic distribution techniques. Like Strong, Rostoker already discloses a server 102 for publishing and distributing applications to businesses, such as business 108, as I discussed above for element [1B]. *See* Section IX.A.3; EX1005, ¶¶21-22, FIG. 1.

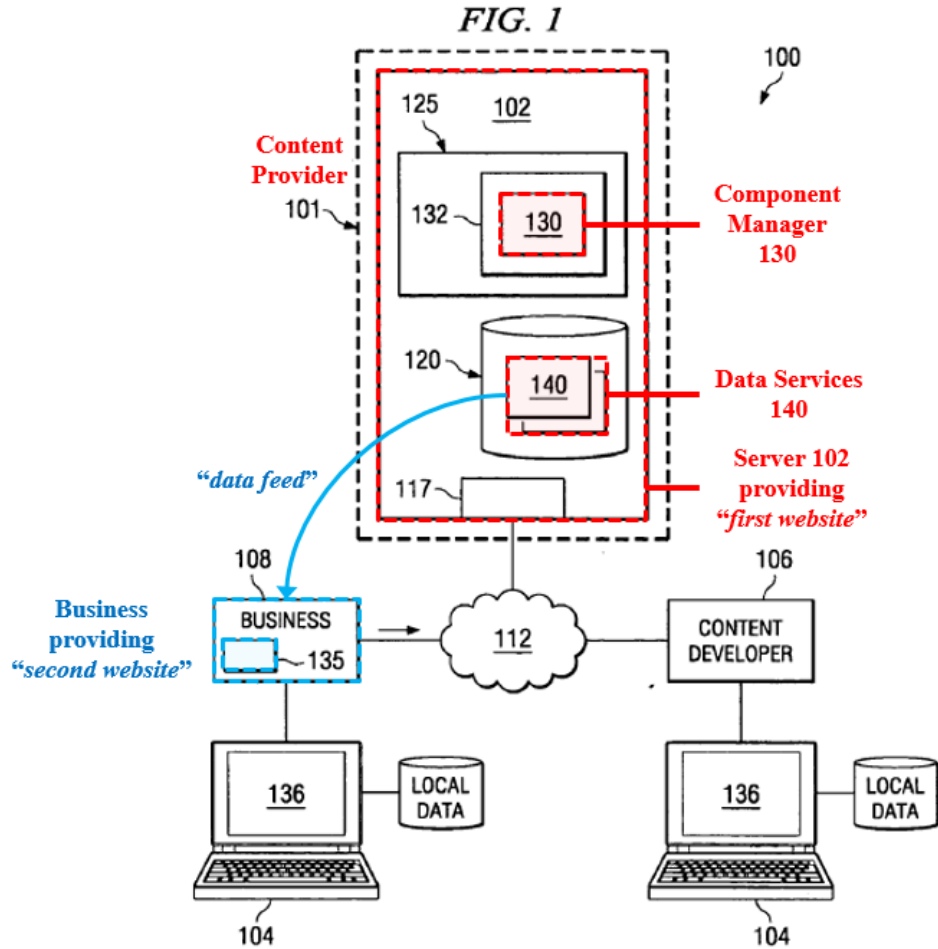
238. Moreover, because server 102 provides Rostoker's "modeling environment," Rostoker's system is already configured to detect modifications to applications as they are developed. EX1005, ¶¶19, 40. For example, as I discuss above, when an "*interactive chart*" is selected for inclusion in an application, Rostoker's system detects the selection and modifies the application to include the new chart. EX1005, ¶¶59, 61, FIGs. 18, 19 (showing selection and addition of chart "Open Remediation Aging" to an application); *see* Sections IX.A.2, IX.A.6. Thus, given that Rostoker's system is already configured to detect modifications to an application, a POSA would have had a reasonable expectation of success publishing/deploying an application upon detection of a modification.

239. Therefore, in my opinion, Rostoker in view of Strong renders obvious “*wherein the at least one interactive chart is automatically displayed on the second website . . . upon the dragging and dropping selection.*”

240. Additionally, I understand that in the parallel district court proceeding, iCharts amended its infringement contentions and asserted that the “*interactive chart is automatically displayed on the second website*” after clicking a “Publish” button. EX1058, 0061 (“The Tableau Accused Products allow ‘publishing’ charts”; “The interactive chart created in the example below can be published by clicking on the ‘Publish’ button . . .”); 0079 (asserting that after an alleged “Drag & Drop” operation, “[t]he interactive chart created in the example above can be published by clicking on the ‘Publish’ button.”). In other words, I understand iCharts to contend that displaying the interactive chart is triggered by clicking a “Publish” button. To the extent such operation satisfies the claim language, Rostoker alone teaches “*wherein the at least one interactive chart is automatically displayed on the second website . . . upon the dragging and dropping selection*” under iCharts’ interpretation. As I explain above, Rostoker discloses the same operation that iCharts appears to accuse in its amended infringement contentions: after dragging and dropping the “*interactive chart*” and clicking a publish button, “*the at least one interactive chart is automatically displayed on the*

second website.” I explained this operation of Rostoker with respect to elements [1B] and [1E]. *See* Sections IX.A.3, IX.A.6.

241. It is also my opinion that Rostoker discloses that “*the at least one interactive chart is automatically displayed on the second website **using a data feed from the first website to the second website.***” As I discuss for element [1A], Rostoker’s content provider 101 and server 102 provide a “*first website*” from which a user may select and create applications. *See* Section IX.A.2; EX1005, ¶¶8, 18, 29, 67. And, as I discuss for element [1E], server 102 provides access to data services 140 through the same “*first website,*” which are “stored in one or more logical or physical repositories” within server 102. *See* Section IX.A.6; EX1005, ¶¶19, 28 (“[T]he component manager 130 is a hosted solution on a remote server 102 providing an easy-to-use interface for disparate data services 140 to at least one business 108 or user”), 32, 39.



EX1005, FIG. 1 (annotated).

242. Annotated Figure 1 of Rostoker above illustrates data services 140 stored on server 102, which provide a “data feed” to business 108 for displaying the interactive chart on the website of the business 108. Therefore, a POSA would have therefore understood that applications deployed to business 108 (providing “the second website”) would rely on a “data feed” to associated data services 140 provided by server 102 (providing “the first website”). Such a “data feed” would allow Rostoker’s charts to act as “[r]eal-time dashboards,” displaying “the current

state of the data.” EX1005, ¶39 (“**Real-time dashboards** often provide table and graph information on the **current state of the data, which may be supplemented by data services 140.**”). Thus, it is my opinion that Rostoker teaches “*the at least one interactive chart is automatically displayed on the second website using a data feed from the first website to the second website,*” as illustrated in the above annotated version of Figure 1.

243. Accordingly, it is my opinion that Rostoker teaches element [1F].

B. Dependent Claim 2

244. Claim 2 depends from claim 1 and recites “*wherein the web widget is a flash widget.*” In my opinion, Rostoker, alone or in view of Strong, teaches this limitation for the reasons I discuss for elements [1B] and [1D] of claim 1. *See* Sections IX.A.3, IX.A.5.

C. Dependent Claim 3

245. Claim 3 depends from claim 1 and recites “*displaying one or more of the plurality of interactive charts in the chart box on the second website.*” In my opinion Rostoker, alone or in view of Strong, teaches this limitation for the reasons I discuss for element [1B] of claim 1. *See* Section IX.A.3.

D. Dependent Claim 4

246. Claim 4 depends from claim 1 and recites “*displaying a plurality of interactive charts embedded in a plurality of chart boxes on the second website.*”

In my opinion, Rostoker, alone or in view of Strong, teaches this limitation for the reasons discussed for element [1B] of claim 1. *See* Section IX.A.3.

X. Ground 2: Rostoker and Strong in view of Jou renders obvious claims 1-4.

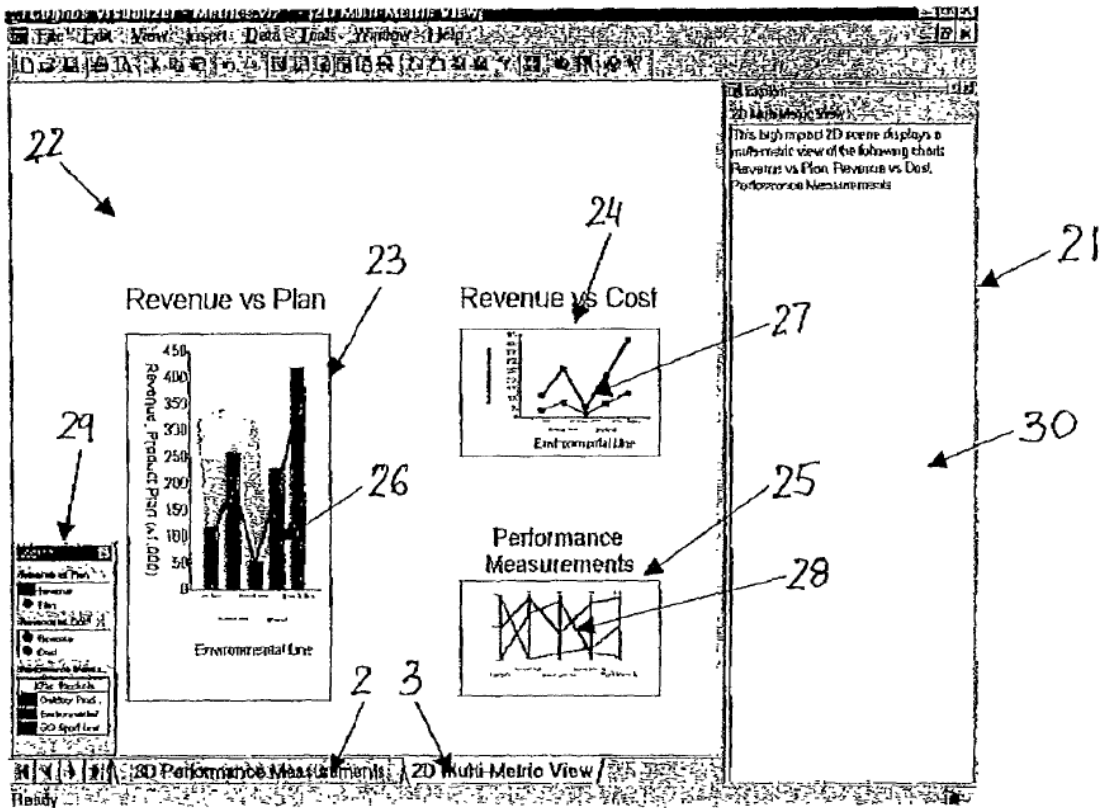
247. The combination of Rostoker, Strong, and Jou teaches or suggests every element in claims 1-4.

248. As I discuss with respect to Ground 1, it is my opinion that Rostoker discloses “*at least one **interactive chart**,*” as recited in element [1A] of claim 1, and “*wherein the at least one interactive chart, when displayed on the second website in the absence of the generation interface, enables a user to a) render the chart, b) obtain input from a user and c) rerender the chart using the input thereby enabling the interactive chart to retain interactivity,*” as recited in element [1D]. *See* Sections IX.A.2, IX.A.5. However, to the extent argued that Rostoker does not expressly disclose the recited “*interactive*” features described in the ’000 patent, Jou discloses such “*interactive*” features, which a POSA would have found useful with Rostoker’s charts.

249. Like Rostoker, Jou “provides an interactive system for visualizing business data.” EX1007, ¶10. Jou describes that “[a] visualization . . . can be built in a number of manners, such as starting from a blank visualization and adding manually various visualization elements, in any order, or importing them from an existing visualization.” EX1007, ¶13. Jou discloses “provid[ing] the user with a

large variety of predefined types of two- and three-dimensional charts, such as line, bar, surface, pie, swatch, and map charts,” as illustrated in Figure 2B below.

EX1007, ¶¶14, 36. Jou also discloses that a user can interact with its charts using “standard input and output devices, such as a mouse and a keyboard.” EX1007, ¶34.



EX1007, FIG. 2B.

250. Jou’s charts include multiple “interactive” features. For example, Jou discloses various “filter[s],” to filter data included in a visualization. EX1007, ¶¶38-41 (“In the method and system according to the invention, **data to be included in a visualization may be filtered**, i.e., limited to a subset of data

available from a given source.”). For instance, Jou discloses “check filter[s],” “radio filter[s],” and “range filter[s]” that allow a user to select data visible in a chart. EX1007, ¶¶38-41.

251. Jou also discloses drill-down features, “allow[ing] **drilling on the dimension axis of charts**, by moving either up or down the hierarchy of categories of a dimension . . . to obtain a more (down) or less (up) detailed view of the data.” EX1007, ¶44. “[T]his is achieved by double-clicking an axis label, with the direction of the drilling indicated by the form of the cursor.” EX1007, ¶44. Jou also discloses a zoom feature for charts. For example, “the user can obtain a closer area view of a map or layer (**zooming in**), [or] obtain a wider area view of a map or layer (**zooming out**)” EX1007, ¶49.

252. Jou’s “*interactive*” features are consistent with the ’000 patent’s description of “interactive features that allow a user to interact with the chart,” providing examples such as “a zoom feature, a data mouse-over pop-up window, a data hide/share feature, an audio playback feature, a drill down feature, a data gliding feature and the like.” EX1001, 9:9-16. Therefore, in my opinion, Jou expressly discloses “*at least one interactive chart*” (element [1A]), “*wherein the at least one interactive chart, when displayed . . . in the absence of the generation interface, enables a user to a) render the chart, b) obtain input from a user and c)*

rerender the chart using the input thereby enabling the interactive chart to retain interactivity” (element [1D]).

253. In my opinion, a POSA would have been motivated to incorporate Jou’s additional “*interactive*” features into Rostoker-Strong’s system to enable users to better visualize and explore business data. As Jou explains, “[t]he vast amounts of warehoused or otherwise collected business data would be useless without software tools for its analysis.” EX1007, ¶8. “[T]ools for visualizing business data are of particular importance,” and “[b]y interacting with person’s sight, they became means of choice for consolidating and presenting vast amounts of complex information, in the manner which facilitates pattern recognition in the presented data and invites data exploration.” EX1007, ¶9. Rostoker is specifically concerned with development of “business application[s].” EX1005, ¶4. Therefore, it is my opinion that a POSA would have understood that Jou’s data filtering, drill-down, and zoom features improve a user’s ability to parse and visualize a large amount of business data, supporting Rostoker’s goal of providing “[r]eal-time dashboards.” EX1005, ¶39.

254. In my opinion, a POSA also would have had a reasonable expectation of success incorporating Jou’s additional interactive features into Rostoker-Strong’s system because Flash and Java enable the types of interactions discussed by Jou. EX1005, ¶50, FIG. 2A. In fact, Rostoker already discloses other interactive

features that are executable within a “Flash runtime” or “Java runtime,” as discussed for elements [1A] and [1B] in Ground 1. And it was well known that these technologies supported interactivity and dynamic modification of graphical objects during runtime. *See, e.g.*, EX1042, 0010 (“Almost all of the properties of Macromedia Flash’s intrinsic objects (textfields, graphics, movie clips) can be dynamically modified at runtime using ActionScript. This makes it easy to create highly customizable and adaptable interfaces.”). Therefore, it is my opinion that a POSA would not have experienced any unreasonable technical hurdles in providing Jou’s interactive features within Rostoker’s system, and the combination would yield predictable results (e.g., incorporation of additional interactive chart features into Rostoker’s existing system).

255. For these reasons, and the reasons discussed in Ground 1, the combination of Rostoker, Strong, and Jou also renders obvious claims 1-4.

XI. Ground 3: Rostoker and Strong in view of Clark renders obvious claims 1-4.

256. The combination of Rostoker, Strong, and Clark teaches or suggests every element in claims 1-4.

257. As I discuss for Ground 1, it is my opinion that Rostoker, alone or in view of Strong, renders obvious “*displaying, in response to the selection the at least one interactive chart embedded in a chart box on a second website,*” as recited in element [1B]. However, to the extent it is argued that Rostoker and

Strong do not expressly disclose that its “*interactive chart*” is displayed “*on a second website*,” it is my opinion that Clark also provides this teaching.

258. Clark is directed to “business data visualization and publishing systems.” EX1010, ¶3. Like Rostoker, Clark provides a system “that generates rich charts, graphs, and maps for enhanced visualization,” allowing users “**to build web sites, portals, and dashboard[s]** using a drag-n-drop interface.” EX1010, ¶¶28-29. Clark’s “data visualization and publishing system 100 provides a streamlined and intuitive interface for **users with ordinary browsers to rapidly access data sources, visualize information, and publish results.**” EX1010, ¶41. As Clark explains:

An intuitive user browser interface at each user is used to cause informational data to be extracted from business applications, databases, and other sources. Data is transformed, merged, and formatted, using a menu-driven graphical user interface. Standalone dashboards, portals, web pages, and components to be integrated into groupware or enterprise portals can be published.

EX1010, ¶41.

259. For example, “[t]o generate attractive charts and map graphics, The B3 system 200 can use the POPCHART and Optimap products sold by Corda.” EX1010, ¶52; *see also* EX1010, ¶47 (“POPCHART is described by Corda as a server-based suite of tools for creating sophisticated interactive charts and graphs

in Macromedia® FLASH, SVG, PNG, JPEG, PDF, EPS, TIFF, WBMP, etc.”).

Therefore, similar to Rostoker, Clark provides a “web-based” interface for “creating and publishing business dashboards” for display within a web browser.

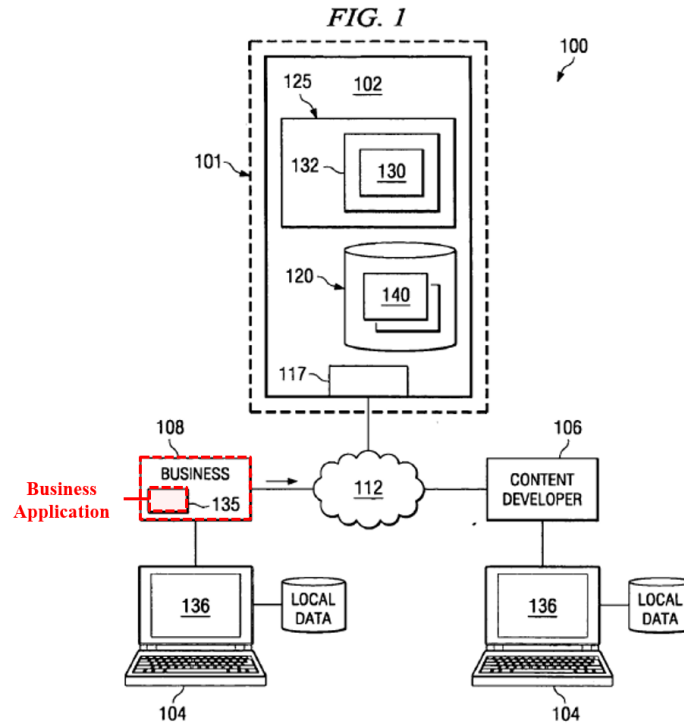
EX1010, ¶¶41, 52.

260. In this manner, Clark discloses that users can “publish” charts for display on “*a second website*,” e.g., within an “enterprise portal”: “The system 200 can **publish standalone dashboards, portals, or web pages**, as well as components **to be integrated into groupware or enterprise portals**.” EX1010, ¶49; *see also* EX1010, ¶117 (“Publishing—**create web pages, dashboards, and portals, or publish content to groupware and external portals**”). A POSA would have understood Clark’s “enterprise portal” to be a “*website*.” For example, the Microsoft Computer Dictionary from the relevant time period defines “portal” as “[a] **Web site** that serves as a gateway to the Internet,” and an “enterprise information portal” as “[a] **portal** or gateway that allows internal and external users in a business or enterprise to **access information from intranets**, extranets, and the Internet for business needs.” EX1066 (Microsoft Computer Dictionary), 11, 12.

261. Clark also describes its “enterprise portal” which resides, for example, “on a company’s intranet,” separate from the “web-based” interface for “creating and publishing business dashboards.” EX1010, ¶¶52, 137 (“[A dashboard] could

also be **published on the company's intranet** to enhance collaboration between employees"). For example, Clark describes that its "web layer" includes "core" functions such as "providing the user services to publish the charts and portal pages." EX1010, ¶107. Therefore, it is my opinion that Clark teaches publishing and displaying interactive charts "*on a second website.*"

262. A POSA would have been motivated to incorporate Clark's techniques of publishing interactive charts to a business's intranet to enable Rostoker's applications to be "integrated into . . . enterprise portals," as taught by Clark. EX1010, ¶49. Rostoker is specifically directed to "configuring an analytical application or dashboard to create a new business application and, thus, quickly respond to changing business needs with a low associated cost." EX1005, ¶5. In fact, Rostoker teaches that its system may be used "in a dedicated enterprise environment" and illustrates in Figure 1 a business 108 storing "composite applications 135 (or xApps)." EX1005, ¶¶21-22, 28, FIG. 1. Clark's techniques would therefore enable a business to internally host and present, through a business's enterprise network (e.g., a local intranet), business applications created using Rostoker-Strong's system.



EX1005, FIG. 1 (annotated).

263. A POSA also would have had a reasonable expectation of success incorporating Clark’s techniques because such incorporation would amount to nothing more than incorporating well-known content-publishing techniques into Rostoker-Strong’s known system. Both Rostoker and Clark teach publishing content for execution in a “Flash runtime.” EX1005, ¶50; EX1010, ¶¶47, 52, 115 (“The system can **publish data** as a table **or using a wide-variety of charts**. ... **Charts can be generated as Flash, SVG, JPEG, and PNG.**”). And Rostoker already discloses a “Publish” function that allows a user to “deploy a ready-to-use analytics application,” similar to Clark’s web-based solution for publishing business dashboards. EX1005, ¶¶27, 71, FIGs. 56, 57; EX1010, ¶¶115, 125.

264. Moreover, techniques for providing business applications through an enterprise web server were well known at the time of the '000 patent. For example, Nikolaidou describes that “Enterprise information systems (EISs) consist of a combination of interrelated **Intranet-based** and Internet-based applications, built on multi-tiered client-server models.” EX1064, 0001. “The Web platform,” which includes a “web server,” “serves well as **a common access environment for all the applications operating within enterprise Intranet boundaries.**” EX1064, 0001. Therefore, a POSA would not have experienced any unreasonable technical hurdles incorporating Clark’s publishing techniques within Rostoker-Strong’s system, and the combination would yield predictable results (e.g., publishing Rostoker’s interactive charts to a business website).

265. For these reasons, and the reasons I discuss in Ground 1, it is my opinion that the combination of Rostoker and Clark renders obvious claims 1-4.

XII. Ground 4: The QlikView Manuals in view of Bales render obvious claims 1, 3, and 4.

266. The QlikView Manuals in view of Bales teach or suggest every element in claims 1, 3, and 4.

A. A POSA would have been motivated to combine the QlikView Manuals.

267. A POSA would have combined the QlikView Reference Manual (EX1008) and the QlikView Server Manual (EX1009) because both references are

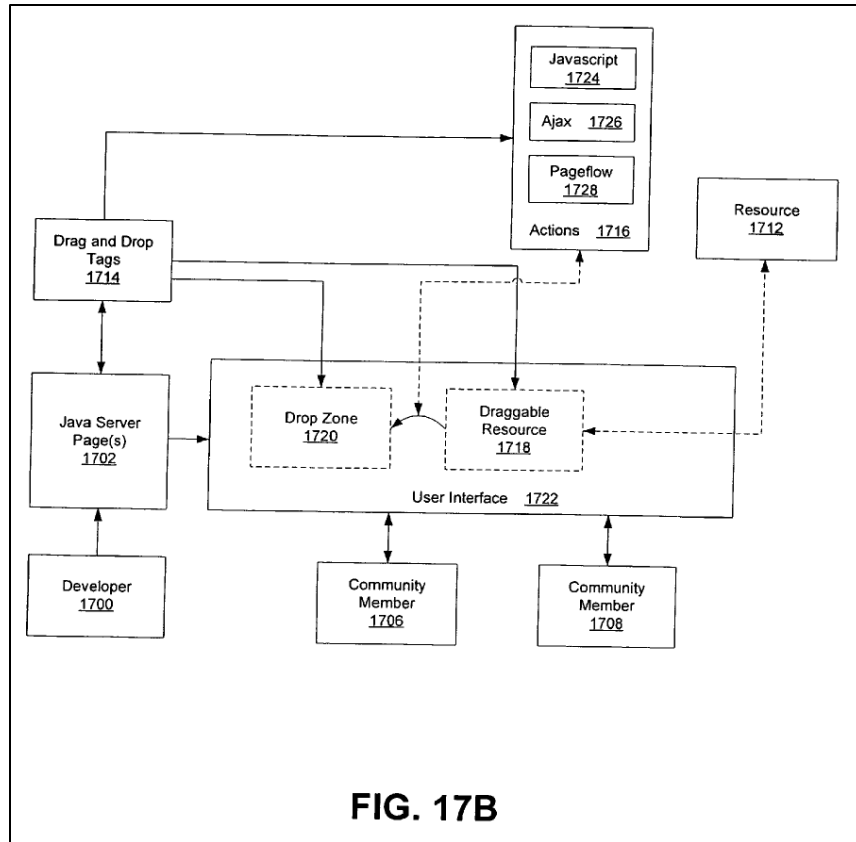
from the same integrated product suite comprising QlikView, QlikView Server, and QlikView Publisher. *See* EX1008, 0019; EX1015-17. Both describe the same product version (8.5). EX1008, 0001; EX1009, 0001. Additionally, if a POSA had any questions about how QlikView worked with QlikView Server, the QlikView Manuals themselves tell the POSA to cross reference one another. EX1008, 0033 (“For more information about the new features in QlikView Server, please refer to the QlikView Server manual.”); EX1009, 0026 (“See reference manual for QlikView”), 0114 (“please consult the QlikView Reference Manual.”), 0141 (“For a detailed explanation of each of the available commands, please refer to the QlikView Reference Manual”), 0172 (“For more information on this, see the QlikView reference manual.”), 0173 (“For more information on how to build such an application, see the QlikView reference manual.”). Accordingly, a POSA would have understood that the QlikView Manuals describe the same system and would have had a reason to interpret the manuals’ teachings collectively.

268. Combining teachings in the QlikView Manuals would have combined prior art elements (QlikView and QlikView Server) according to known methods (server-client interactions) to yield predictable results (generating and publishing interactive charts). A POSA would have also had a reasonable expectation of success combining the teachings in the QlikView Manuals because the teachings were in fact combined to make QlikView work as a real system.

B. A POSA would have been motivated to combine the QlikView Manuals and Bales.

269. The QlikView Manuals explain how developers can publish QlikView documents and objects (e.g., charts) online. EX1009, 0134-36, 0144-49, 0156-58. For example, the QlikView Manuals detail how a developer can create QlikView Java Objects and publish them within webpages. EX1009, 0134-36. However, a POSA would have understood that additional functionality to dynamically update published charts would further QlikView's "major design goals" of "enhanc[ing] user collaboration." EX1009, 0013. A POSA would have therefore searched for references explaining how to update published content dynamically. This would have led the POSA to Bales, which discloses a system that updates content on a website dynamically upon drag-and-drop operations.

270. Shown below, Bales discloses a system that selects web content by dragging a draggable resource 1716 into a drop zone 1720 (e.g., a folder). EX1006, ¶168. Upon dropping the content into the drop zone, an action 1716 is performed. EX1006, ¶168. "This action can be specified in the drag and drop tag and can take the form of a Javascriptfunction [sic] (1724), an AJAX call (1726) or a page flow action (1728)." EX1006, ¶168.



EX1006, FIG. 17B.

271. Additionally, “a dynamic content call can be made to a remote server” EX1006, ¶168. As Bales explains, the dynamic content calls can update a second website automatically using AJAX. EX1006, ¶¶138, 147.

272. A POSA would have been motivated to use Bales’s drag-and-drop approach because it is “a more abstract, dynamic and user-friendly approach, one that would provide a more fluid and adaptable system of collaboration between users over a distributed computing environment.” EX1006, ¶21. As Bales explains, “[t]he drag and drop library can provide for a highly user-friendly and fluid interface.” EX1006, ¶161.

273. A POSA would have also been motivated to use Bales's drag-and-drop approach to update the second website dynamically. As Bales explains, updating a portion of a website instead of the entire browser allows for a much more responsive user interface and data exchange:

For example, a stock trading value within a portlet can be updated via the dynamic content tags feature, without re-posting the entire browser or even the portlet. Full pages, portions of pages, tables, portions of tables as well as other pieces and segments can be updated dynamically. Such granularity of communication allows for **a much more responsive user interface and data exchange.**

EX1006, ¶125.

274. A POSA would have also been motivated to use Bales's drag-and-drop approach to reduce server workload. As Bales explains, "[a]nother advantage can be reduced serverworkload [sic] because the server is not burdened with re-running security on every portlet, re-rendering all HTML for the entire page and re-downloading all of the Javascript everytime [sic] that a communication occurs."

EX1006, ¶125. With AJAX, the system can update only the portion of the website when the content changes instead of the entire website. "[A] granular and preemptive updating process is achieved, yet requiring very little or no user interaction." EX1006, ¶147.

275. Adding Bales's drag-and-drop approach to the QlikView Manuals would have combined prior art elements (JavaScript/AJAX, drag-and-drop, drop zones) according to known methods (basic web development) to yield predictable results (updating content on a website). By early 2009, JavaScript and AJAX were well-known web development tools that were used commonly to publish and update content online. Drag-and-drop was also ubiquitous in the industry. Because JavaScript, AJAX, and drag-and-drop were well known, a POSA would have known how to predictably implement them within a data visualization tool.

276. Adding Bales's drag-and-drop approach to QlikView would have used a known technique (drag-and-drop) to improve similar devices (AJAX/Java-based website development tools) in the same way. QlikView and Bales both use JavaScript and AJAX to publish and update content online. EX1009, 0134-36, 0144-49, 0156-58; EX1006, ¶¶97-99, 122, 124-25, 131, 147, 158, 162-68, 176. Because QlikView (like Bales) also stores its website content server side, a POSA would have known how to update the published content automatically in the same way. As Bales explains, clients would ping the server every so often to update the published content automatically. EX1006, ¶¶139-49.

277. Additionally, a POSA would have had a reasonable expectation of success because JavaScript, AJAX, and drag-and-drop were all mature technologies before 2009. There would have been no technical hurdles that a

POSA would have been unable to overcome. Indeed, Bales provides ample technical details (including sample code) to predictably implement this known functionality. EX1006, ¶¶131, 162-63, 166.

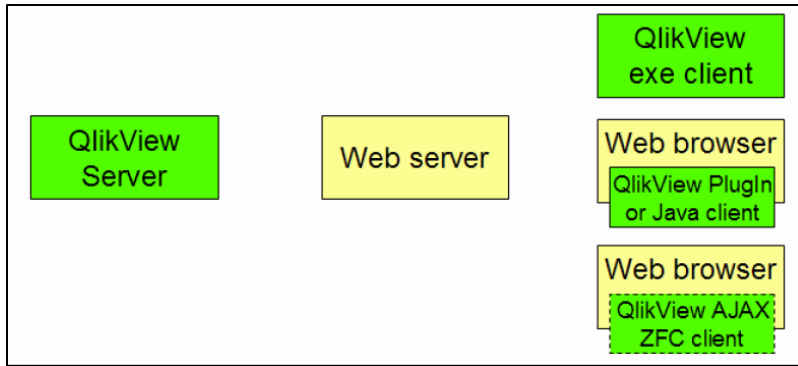
C. Independent Claim 1

278. The QlikView Manuals in view of Bales teach or suggest every element in claim 1.

1. [1P] “A computer-implemented method”

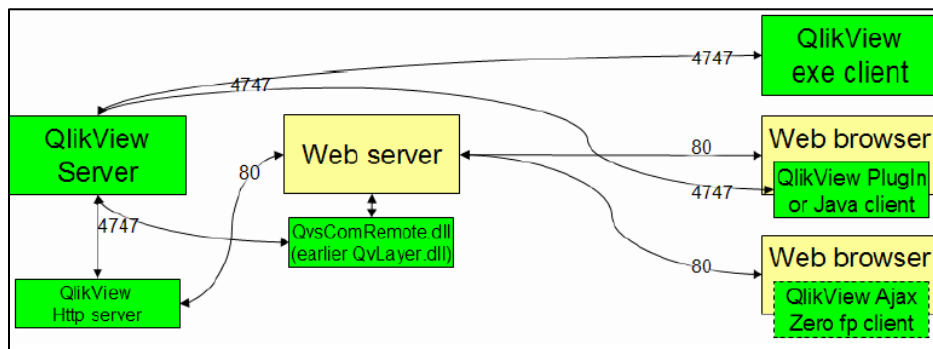
279. Claim 1 recites “[a] *computer-implemented method*.” To the extent the preamble is determined to be limiting, the QlikView Manuals disclose the preamble.

280. The QlikView Manuals teach the preamble. Particularly, the QlikView Manuals describe methods implemented on a computer system having a QlikView Server, a Web Server, and a variety of QlikView Clients. EX1009, 0095. “This interaction can potentially involve multiple computers and multiple network connections.” EX1009, 0095; EX1008, 1326-27 (system requirements). For example, QlikView can run on a computer with Microsoft Windows, Microsoft Windows Server, Microsoft Windows XP, or Microsoft Vista. EX1008, 1327.



EX1009, 0095.

281. The figure below illustrates network connections within the QlikView system. For example, Windows Clients running QlikView directly communicate with the QlikView Server, whereas Java and AJAX clients use a web server to facilitate communication with the QlikView Server. EX1009, 0098.



EX1009, 0098.

282. For at least these reasons, the QlikView Manuals disclose [1P].

2. [1A] “receiving a selection, on a first website, of at least one interactive chart from a plurality of interactive charts”

283. Claim 1 recites “receiving a selection, on a first website, of at least one interactive chart from a plurality of interactive charts.” The QlikView Manuals disclose this limitation.

284. Shown below, the QlikView Manuals disclose “*a first website*” that has a plurality of sheet objects, including charts. The QlikView Manuals state that “[c]harts and tables are sheet objects” EX1008, 0880. These examples illustrate users generating charts using different web clients. EX1009, 0097.

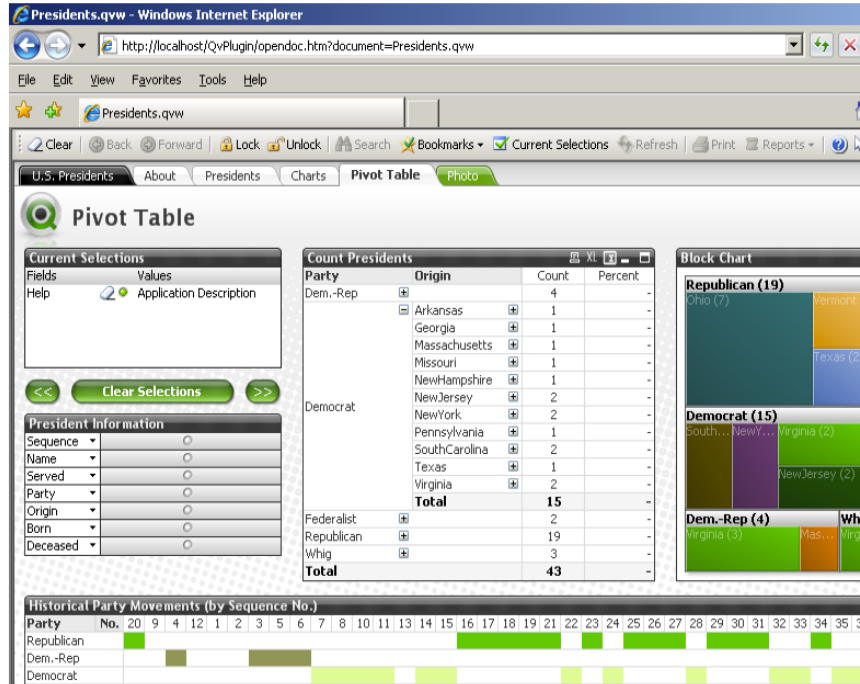
285. For example, the image below depicts “*a first website*” with different types of sheet objects (charts) within a QlikView Java client.



QlikView Java client in Internet Explorer.

EX1009, 0129.

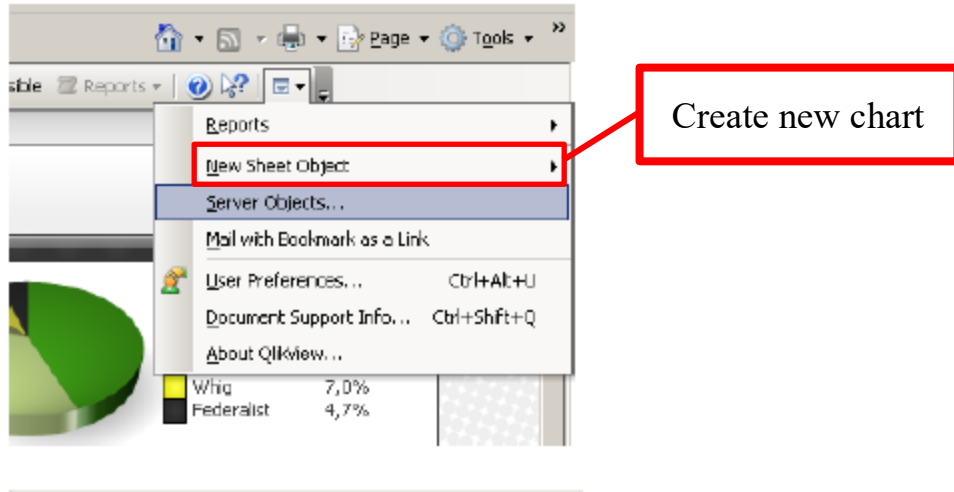
286. Similarly, the screenshot below depicts “*a first website*” with different types of sheet objects (charts) within a plugin client for Internet Explorer.



QlikView plug-in client in Internet Explorer.

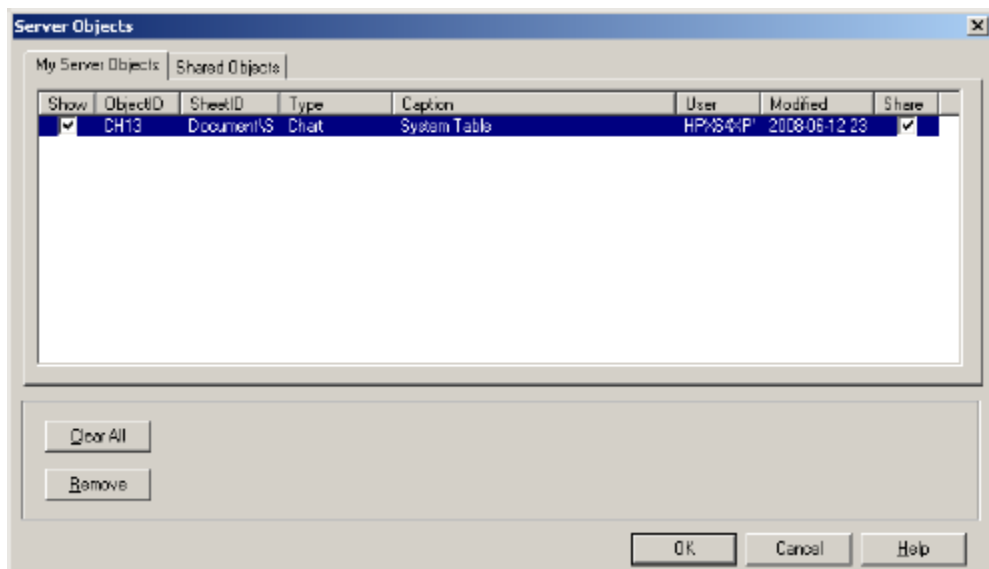
EX1009, 0139.

287. The web clients displaying interactive charts on “*the first website*” can run QlikView Professional and not limited to QlikView Analyzer. EX1009, 0201; *see also* EX1008, 0016 (explaining differences between QlikView Professional and QlikView Analyzer). In QlikView Professional, the user can “create personal sheet objects [e.g., charts] and share them with other users.” EX1008, 0016. This functionality is shown below in the QlikView plug-in for Internet Explorer.



EX1009, 0140.

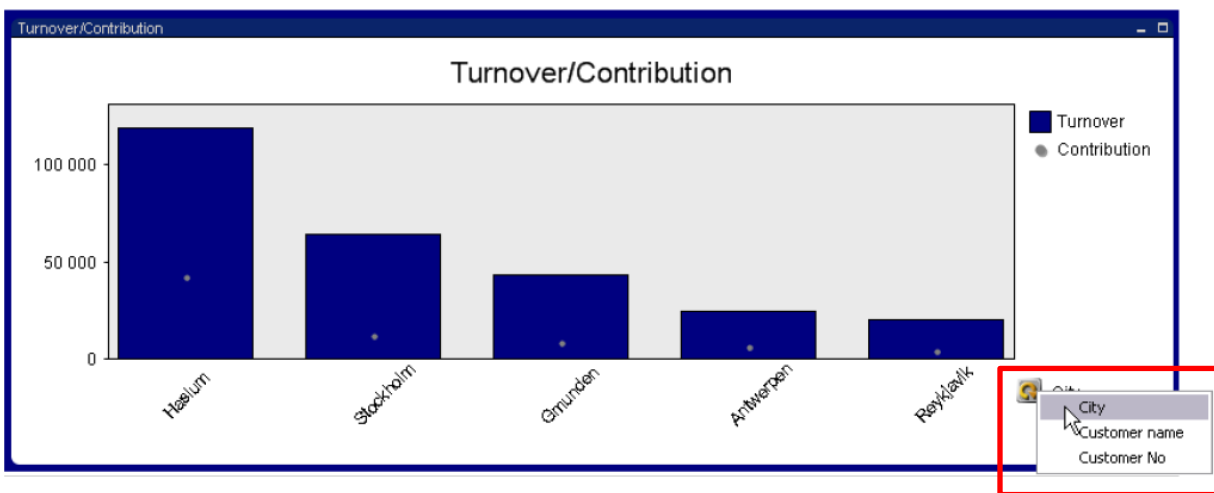
288. The plug-in allows the generated charts to be shared with other users via the “Server Objects” dialog. EX1009, 0140. Shown below, when a user clicks the “Share” checkbox, QlikView “receiv[es] a selection, on a first website, of at least one interactive chart from a plurality of interactive charts.”



EX1009, 0140.

289. The charts on the first website are “*interactive charts*.” EX1008, 0013 (charts “dynamic”). For example, “[s]elections in field data can be made directly in most of QlikView’s charts and tables by means of clicking or painting with the mouse.” EX1008, 0154-55, 0881. Selections (e.g., interactions) are made with the plot area, chart legend, dimension axes, and expression axes. EX1008, 0154. QlikView further provides different interactive functions based on the type of chart. For example, in a scatter plot the user can select an individual data point whereas in a pie chart the user can select an individual pie slice. EX1008, 0154-55.

290. The QlikView Manuals also describe a drill down feature, where a user can select a field to update chart dimensions based on the selected field. EX1008, 1306. Furthermore, as highlighted below, the user can interact with a cycle icon in a chart to change the field (e.g., “City,” “Customer name,” “Customer No”) used as the chart’s dimension. EX1008, 1307.



EX1008, 1307 (annotated).

291. The charts also support pop-up labels showing data point values when the user moves the cursor over the data point. EX1008, 0919. Other interactive features include a “Fast Type Change” allowing the user to select and convert the chart to a different type, interactive search feedback, sort indicators within tables, and drop-down select within tables. EX1008, 0668-69; EX1009 0016-17.

292. The QlikView Manuals also disclose interactive slider objects and buttons. EX1008, 0036. For example, if a chart includes an animation, an animation bar is displayed in the chart area, including a play button and a pause button. EX1008, 0899.

293. As a POSA would have understood, the interactive features for QlikView charts are mutually or reciprocally active and provide real-time feedback to users to further modify the charts.

294. The interactive features described in the QlikView Manuals are similar to those described in the '000 patent. Specifically, the '000 patent discloses interactive features including “a zoom feature, a data mouse-over pop-up window, a data hide/share feature, an audio playback feature, a drill down feature, a data gliding feature and the like.” EX1001, 9:13-16. The '000 patent states that these interactive features allow the user to modify or change a view of the chart, or learn more about the chart data. EX1001, 9:9-13. As discussed above, QlikView’s interactive options provide similar “interactive features.”

295. Once generated, QlikView can “*receiv[e] a selection*” of one of the charts. For example, a user can select a chart by right clicking the chart to open the object menu. EX1008, 0090, 0624, 0668, 1014. The object menu displays options based on the type of chart selected. EX1008, 0668. For example, if the user created a pie chart, they can right click the pie chart, open the object menu, and update the pie chart’s style, as depicted below. EX1008, 1014-18.



Figure 37. The Style page for pie charts in the Chart properties dialog

EX1008, 1018.

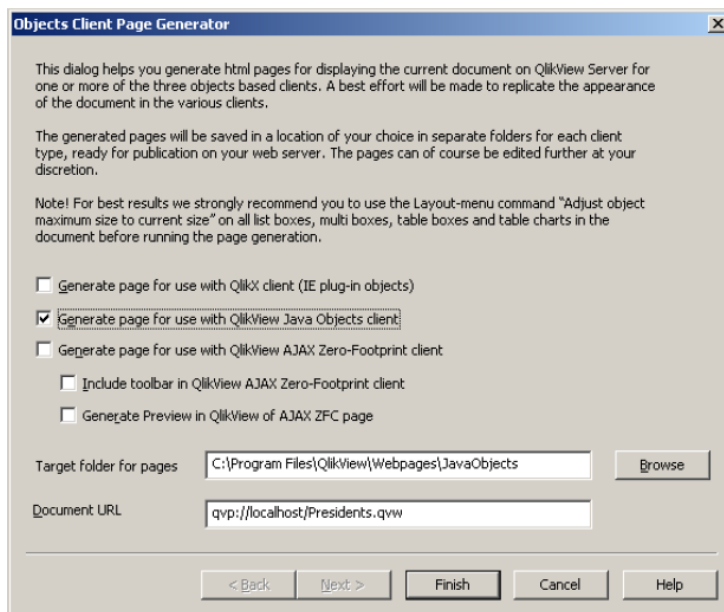
296. QlikView also allows users to select individual charts to publish in an online HTML environment:

The QlikView Objects Clients (Java Objects, QlikX, and AJAX) are based on the concept of placing individual sheet objects [e.g., charts] from a QlikView document in an HTML environment. This gives the

web designer the freedom of placing single objects in arbitrary places on a web site and integrating them with text and other HTML code.

EX1009, 0009-010, 0134-36, 0156-57; EX1008, 0807-08 (publishing sheet objects using the client page generator).

297. For example, using the Objects Client Page Generator Tool below, the user can create a webpage including a chart as a Java Object. EX1009, 0133.



The Objects Client Page Generator for Java Objects client.

EX1009, 0134.

298. Once created, the Java Object can be embedded within HTML for display on a second webpage. EX1009, 0135-36. As a POSA would have understood, creating and publishing charts to a second website requires Qlikview to “receiv[e] a selection, on a first website, of at least one interactive chart from a plurality of interactive charts.”

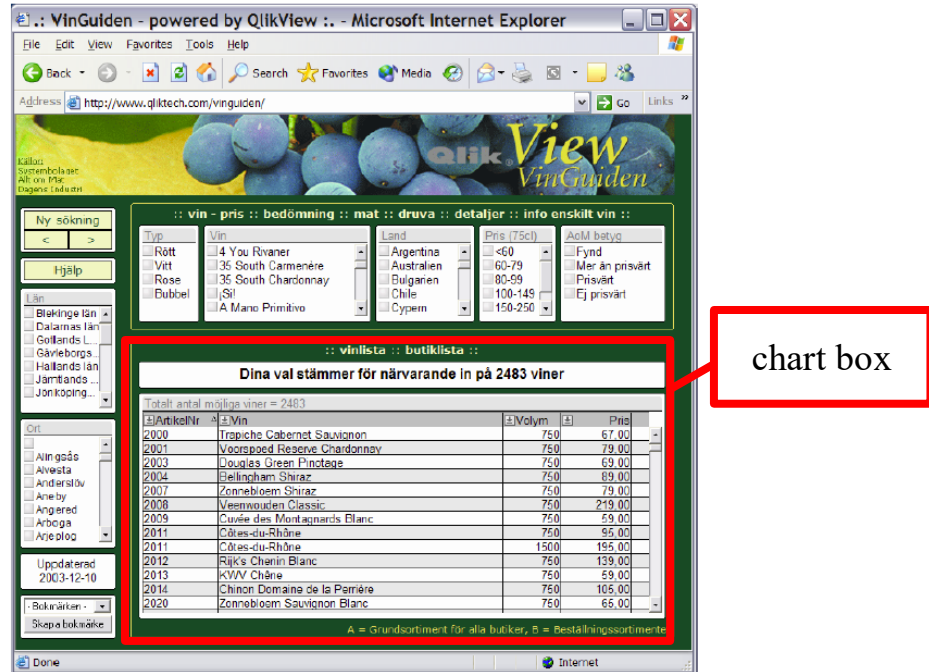
299. For at least these reasons, the QlikView Manuals disclose [1A].

3. [1B] “displaying, in response to the selection the at least one interactive chart embedded in a chart box on a second website, wherein the chart box is a web widget”

300. Claim 1 recites “*displaying, in response to the selection the at least one interactive chart embedded in a chart box on a second website, wherein the chart box is a web widget.*” The QlikView Manuals disclose this limitation.

301. In response to the selection of an interactive chart (*see* element [1.A] above), QlikView displays the chart embedded in a chart box on a second website, as shown below. As The QlikView Manuals explain, “[t]he QlikView Objects Clients (Java Objects, QlikX, and AJAX) are based on the concept of placing individual sheet objects from a QlikView document in an HTML environment.”

EX1009, 0009.



QlikView Java Objects client in Internet Explorer.

EX1009, 0133 (annotated).

302. The second website is different than the first website discussed above for element [1A] because it is accessed by a different client at a different URL. The QlikView Manuals state that “[a]ny combination of client types” are allowed, and that “any QlikView document may be displayed with any client.” EX1009, 0123. Each QlikView client provides a URL defining where the first website is run. For example, as shown above, the “second website” has a different URL than the “first website” in [1A]. Compare EX1009, 0133 (“http://www.qliktech.com/vinguiden”), with EX1009, 0129, 0139 (“http://localhost/QvJava...”/“http://localhost/QvPlugin...”).

303. The chart box is a “*web widget*” because embed code for the chart box is pasted into the second website’s HTML code. Below is sample code for embedding an interactive chart in a chart box. EX1009, 0136. In this example, QlikView objects are embedded into a second website using HTML and a server-side script. The client webpage is connected to the QlikView document on the QlikView Server via a “setDocument” command. EX1009, 0135. Since each chart has a unique identifier, the user can reference the identifier within the HTML to embed the chart within the web page. EX1009, 0136. As a result, the user can use the “addObject” command and the chart identifier to embed the chart in the webpage. EX1009, 0133.

304. For example, the code below defines four chart boxes (two rows <tr>, each row having two cells <td>) using the “addObject” command. A different QlikView object from the QlikView file (qwt_objects.qvw) is embedded in each chart box.

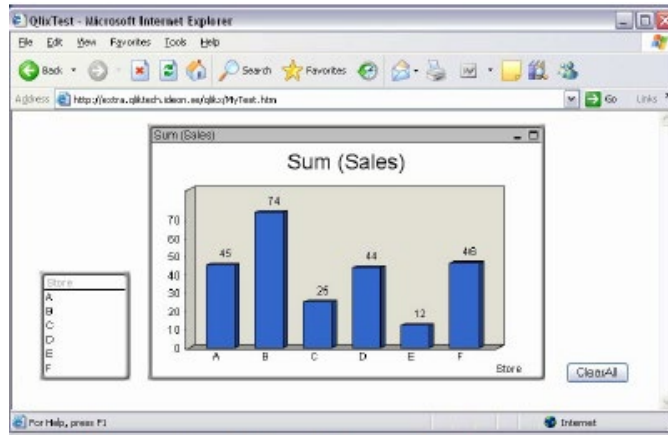
```
<!--#INCLUDE Virtual="/QvJava/Servers/QlikWebObjects/Qlik-  
Web.asp"-->  
<html>  
<head>  
  <title>.: QlikWeb Objects :.</title>  
</head>  
<body>  
  <% setDocument("qwt_objects.qvw") %>  
  <table border="0" cellspacing="0" cellpadding="0">  
    <tr>  
      <td>  
        <% addOb-  
          ject("qwt_objects.qvw", "LB01", 140, 80) %>  
      </td>  
      <td>  
        <% addOb-  
          ject("qwt_objects.qvw", "LB02", 250, 100) %>  
      </td>  
    </tr>  
    <tr>  
      <td>  
        <% addOb-  
          ject("qwt_objects.qvw", "LB03", 140, 140) %>  
      </td>  
      <td>  
        <% addOb-  
          ject("qwt_objects.qvw", "CH01", 250, 150) %>  
      </td>  
    </tr>  
  </table>  
</body>  
</html>
```

EX1009, 0136.

305. As an additional example, the code below is used to generate the displayed list box and chart. EX1009, 0147.

[illegible]

EX1009, 0148-49.



The QlikX example in Microsoft Internet Explorer.

EX1009, 0149.

306. Chart boxes are also defined within a Java applet. As a POSA would have understood, a Java applet is a “web widget.”

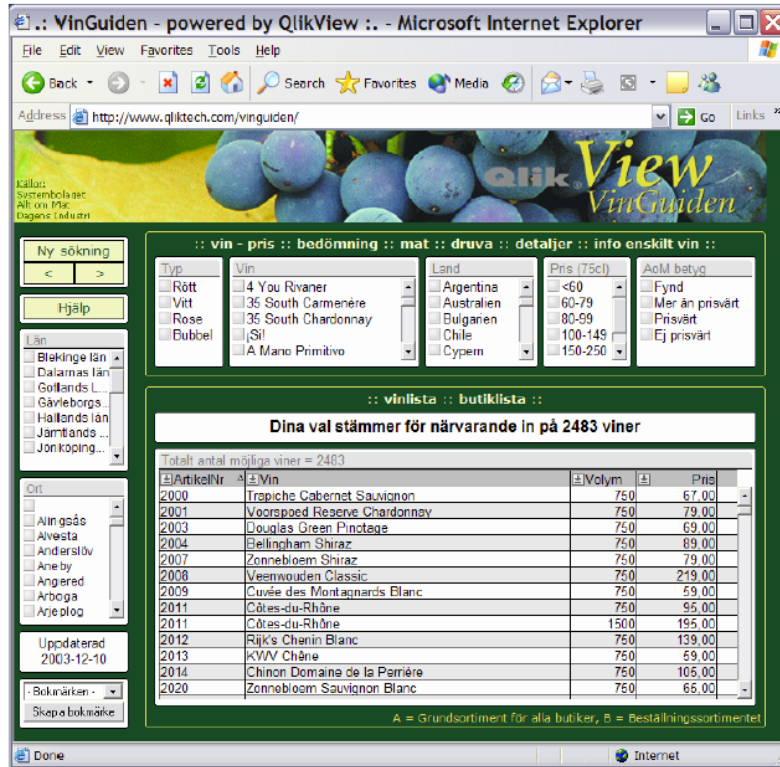
```
<applet code='com.qt.qw.QlikApplet.class'
  archive='/QvJava/QlikView.jar' height=700 width=1000
  MAYSCRIPT>
  <param name='DOCUMENT' value='Presidents.qvw'>
  <param name='PORT' value='4747'>
  <param name='ISAPI' value='/scripts/QVSTunnel.dll'>
  <param name='HEIGHT' value='700'>
  <param name='WIDTH' value='1000'>
  <param name='LANGUAGE' value='English'>
</applet>
```

EX1009, 0130.

307. The QlikView manuals also disclose a “master applet” to handle communication between the QlikView Server and the QlikView Objects. EX1009, 0135. Instead of each QlikView Object communicating with the QlikView Server, a “master applet” handles communications between the QlikView Server, the QlikView file, and the specific QlikView Objects rendered on the webpage.

EX1009, 0135.

308. The published charts remain interactive. For example, shown below, a chart published for a Java Objects client remains interactive.



QlikView Java Objects client in Internet Explorer.

EX1009, 0133.

309. A POSA would have understood that the QlikView Manuals disclose interactive charts that are interactive in the chart box on the second website as recited in claim 1 of the '000 patent also because the POSA would have known how QlikView was actively being used on different web applications. For example, a POSA would have been aware that QlikView was being used on an online Web Selector Tool, as shown below. *See* EX1024-27.



EX1025.

310. As indicated below, the table charts support receiving input from the user. For example, the site instructs the user to “make a selection click on the data value you want to know more about.” EX1025. The table chart further reacts (i.e., is interactive) based on the user input. Referring to the “selection click” above, the instructions state that “[t]he selected data will turn green.” EX1025. In response to

the selection, values that are compatible with the selection will be shown in white, and unrelated values will be shown in gray. EX1025.

To make selections

To make a selection click on the data value you want to know more about. The selected data will turn green. Values compatible with the selection are white and unrelated values are gray. To select more than one item in the same listbox, hold the CTRL key down while selecting additional values.

For example, are you looking for a gift or the perfect wine for dinner? Lets select "white wine" in the "Select Color" list box to the right, and also select "Excellent" in the "Select Value" list box. In the "Select Flavor Options" object, type "citrus" next to the word "find" and press "find". In the "Wine Selection Based on Criteria", you can see that your best selections are:

Herman J. Wiemer Chardonnay, 1999

Antonio Esteves Ferreira Soalheiro Alvarinho, 2002

EX1025.

311. The online interactive chart includes other features such as the ability to “undo, redo, or clear” selections. EX1025. The user can also interact with a “cycle button in a chart to view the data in a different way.” EX1025.

Step Back/Forward, Clear Selections, or Lock & Unlock

To undo, redo, or clear your selections, click on the "Back","Forward", or "Clear Selections" button in the left corner of the page.

Suppose we have changed our mind, and want a fruit flavor instead of a citrus flavor. Click the "back" button, and now type "fruit" next to "find" in the "select flavor options" object.

Cycles

If available, click on a cycle button in a chart to view the data in a different way.



EX1025.

312. The image below shows the layout of the Wine Selector tool, including various elements supporting user interaction.

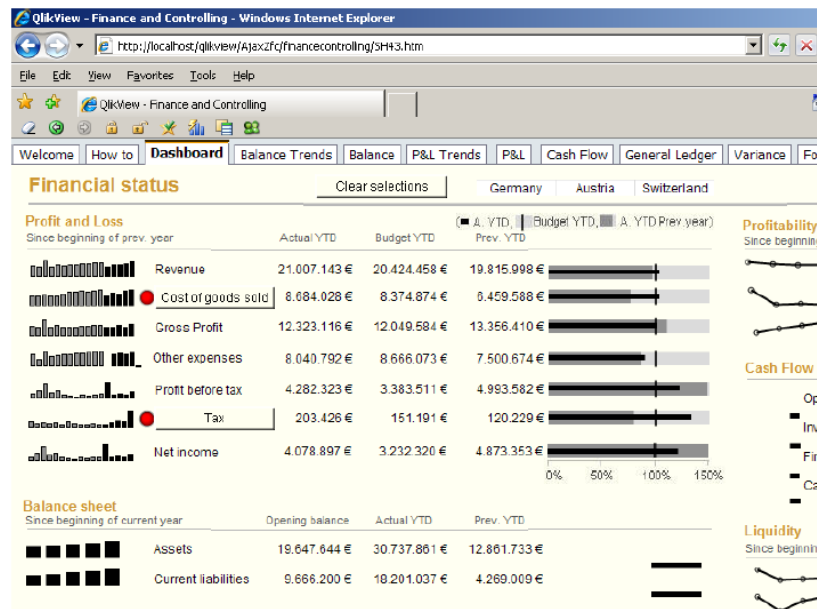
The screenshot displays the 'Wine Selector' web application. At the top, the title 'Wine Selector' is followed by the tagline 'Find Your Perfect Wine for a gift, for a party, or just for dinner.' Below this, there are navigation buttons: 'Back', 'Fwd', and 'Clear Selections'. The main interface is divided into several sections for user input and results. On the left, there are multiple 'Select' dropdown menus: 'Select Value', 'Select Price', 'Select Rating', 'Select Color', 'Select Wine Vintage', 'Select Country', 'Select Flavor Options', and 'Select Varietal'. Each dropdown menu has a 'Loading...' indicator. In the center, there are two more 'Select' dropdown menus: 'Select Food Pairing' and 'Select Year', also with 'Loading...' indicators. On the right, there are two large result panels. The top panel is titled 'Wine Sorted by Flavor' and has a 'Print' button; it contains a 'Loading...' indicator. Below this is a 'Similar' section with a button and the text 'Click to find other wines with this flavor characteristic.' The bottom panel is titled 'Wine Selection Based on Criteria' and also has a 'Print' button; it contains a 'Loading...' indicator. At the bottom of the page, there is a footer with the text 'Need help using the Wine Selector? [Click here](#)', '© QlikTech International. [www.qliktech.com](#)', and 'Powered by QlikView™'.

EX1026.

313. Additional interactive chart features include “drop-down select,” “slider object[s],” “buttons,” “interactive search feedback,” etc. EX1009, 0016-17. The charts also include tool tips, context menus, and LED selection indicators.

EX1008, 0035-36. Furthermore, “[j]ava client users [can also] change certain column sizes in Multi Box and table/chart objects.” EX1009, 0057.

314. The charts published to an AJAX client also remain interactive. EX1009, 0156-57; EX1008, 0807-08. AJAX, “[s]horthand for ‘Asynchronous JavaScript and XML,’ is a development technique for creating **interactive** web applications.” EX1008, 1312. Once published, the QlikView AJAX client is configured to receive user input and update the user interface without refreshing the browser. EX1009, 0155-56. A screenshot of the QlikView AJAX client is shown below.



QlikView AJAX client with the toolbar option

EX1009, 0155.

315. For at least these reasons, the QlikView Manuals disclose [1B].

4. [1C] “wherein the at least one interactive chart is generated on a generation interface configured to allow a user to generate the interactive chart with a chart template from a chart template store”

316. Claim 1 recites “*wherein the at least one interactive chart is generated on a generation interface configured to allow a user to generate the interactive chart with a chart template from a chart template store.*” The QlikView Manuals disclose this limitation.

317. In QlikView, “[c]harts are created by choosing New Sheet Object from the Layout menu, or by pressing the Create Chart button” EX1008, 0667; *see also* EX1009, 0140. “This opens a wizard that will help you create the chart.” EX1008, 0667. The Chart Wizard’s UI (the “*generation interface*”) is below. The Chart Wizard’s UI has multiple tabs (General, Dimensions, Expressions, Sort, Style, Presentation, Areas, Colors, Number, Font, Layout, Caption) that allow users to customize the chart.

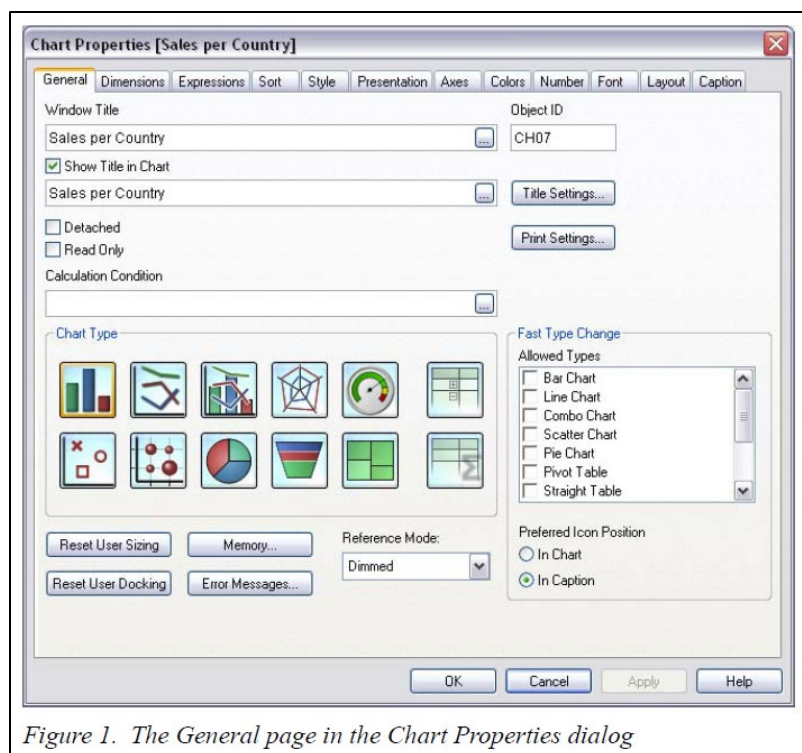


Figure 1. The General page in the Chart Properties dialog

EX1008, 0882; *see also* EX1008, 0085, 0880, 0890, 1104-14 (the Quick Chart Wizard).

318. As discussed for [1A], the generated charts are “*interactive*.” EX1008, 0013 (charts “dynamic”). “Selections in field data can be made directly in most of QlikView’s charts and tables by means of clicking or painting with the mouse.” EX1008, 0154-55, 0881. For example, in a pie chart the user can make a selection by clicking individual pie slices, painting several slices, or clicking information in the chart legend. EX1008, 0154-55. The charts have pop-up labels, slider objects, drill-down, drop-down select, sort indicators, interactive search feedback, buttons, and LED selection indicators. EX1008, 0035-36, 0919. Additionally, the user can move and resize chart components such as the chart’s title, legend, and text.

EX1008, 0669. As a POSA would have understood, these interactive features are mutually or reciprocally active and provide real-time feedback to users to further modify the charts.

319. The Chart Wizard’s UI (the “*generation interface*”) is “*configured to allow a user to generate the interactive chart with a chart template from a chart template store.*” Shown below, the UI’s “Layout” tab allows users to apply a chart theme. EX1008, 0942. A theme is “[a] file containing the properties and look and feel of a document, sheet or object.” EX1008, 1323.

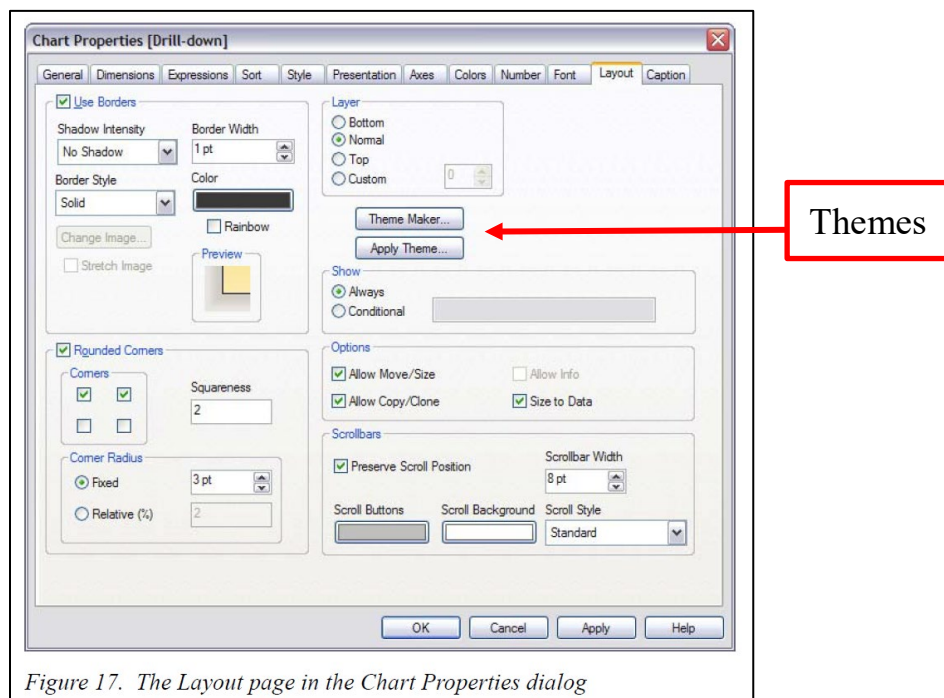


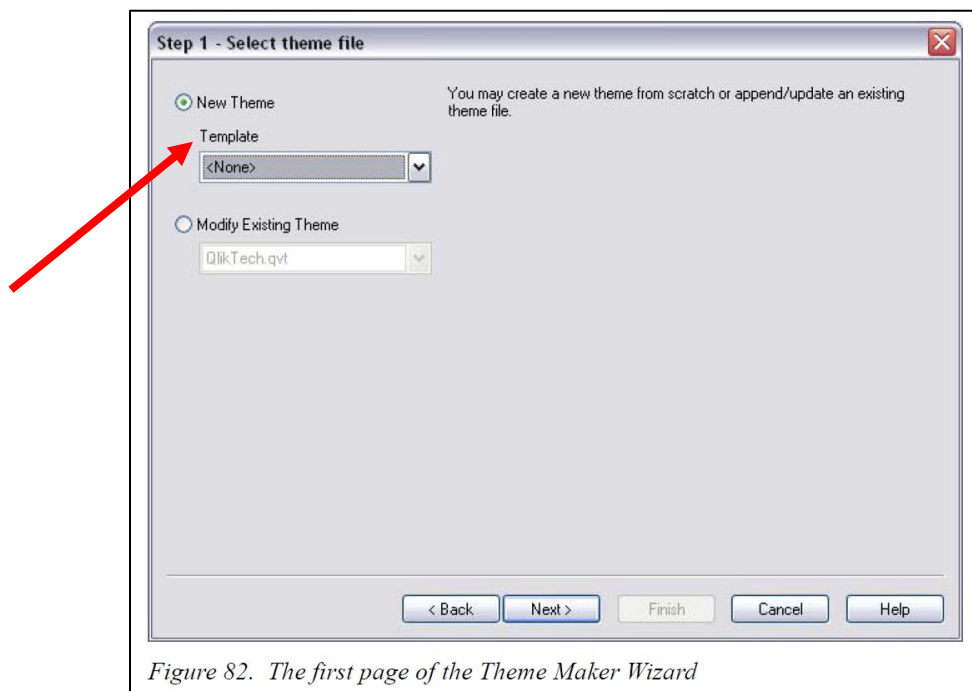
Figure 17. The Layout page in the Chart Properties dialog

EX1008, 0939 (annotated).

320. QlikView’s chart themes are consistent with the ’000 patent’s chart templates, which likewise allow users to create charts with the same “look and feel.” EX1001, 12:31-33 (“Chart templates allow the users to create interactive

charts in the future that have the same look and feel as the generated chart 1700.”).

Indeed, QlikView calls its chart themes “templates.”



EX1008, 0799 (annotated).

321. QlikView themes include sections for the document, sheet, and each type of sheet object (e.g., chart). EX1008, 0795. Each section includes sub-sections for object-specific properties, border and caption properties, and printer setting properties. EX1008, 0795. The user can select the “Theme Maker” button to launch the Theme Maker Wizard and create a new theme, or modify an existing one. EX1008, 0798-99. To modify an existing theme, the Theme Maker Wizard displays existing themes in a default QlikView theme folder, and also provides an option to browse other locations for theme files. EX1008, 0799.

322. The Theme Maker Wizard also allows the user to extract formatting properties from an existing sheet object, sheet, or document. This allows the user to choose properties from the selected object to add to the theme. EX1008, 0800-01. For example, the user can select a slider object and identify properties of the slider object to include in a new theme, such as the background, slider color, button color, and style. EX1008, 0802. The user can then apply the extracted properties to other objects in the sheet such as charts and text objects. EX1008, 0803. As a final step, the user saves the theme and can choose to apply it to the current document or all new documents. EX1008, 0804.

323. The selected theme (the “*chart template*”) is “*from a chart template store*” because the chart themes/templates are in a dedicated theme folder:

Theme files are coded in xml and normally kept in a special QlikView theme folder under Windows’ Application Data folder for the active user.

EX1008, 0795; *see also* EX1008, 0799 (“The drop-down will list all existing themes in your default QlikView theme folder.”).

324. For at least these reasons, the QlikView Manuals disclose [1C].

5. [1D] “wherein the at least one interactive chart, when displayed on the second website in the absence of the generation interface, enables a user to a) render the chart, b) obtain input from a user and c) rerender the chart using the input thereby enabling the interactive chart to retain interactivity”

325. Claim 1 recites “*wherein the at least one interactive chart, when displayed on the second website in the absence of the generation interface, enables a user to a) render the chart, b) obtain input from a user and c) rerender the chart using the input thereby enabling the interactive chart to retain interactivity.*” The QlikView Manuals disclose this limitation.

326. As I explained above in element [1B], the QlikView Manuals disclose a system that displays the interactive chart on a second website. This interactive chart is displayed “*in the absence of the generation interface*” because the second website is separate from the first website. For example, the QlikView Manuals disclose displaying interactive charts via a Java Client, where a Java applet is integrated with a web browser. EX1009, 0129. The QlikView Manuals also disclose displaying interactive charts via a Java Object Client, where each chart is implemented as a Java applet within a web page. EX1009, 0133.

327. “Java programs [] are self-contained applications written in Java.” EX1009, 0199. Java applets are Java programs executed within a Java Runtime Environment (JRE). EX1009, 0199. The JRE provides a sandbox for the applet.

EX1009, 0199. As a result, the Java applets are restricted to the JRE and cannot access resources outside the JRE. EX1009, 0199.

328. This functionality allows the interactive chart to be used by a QlikView client in the absence of the generation interface. For example, a chart can be created using QlikView Developer and stored on a QlikView Server. EX1008, 0016. Subsequently, an end user with QlikView Analyzer or QlikView Professional installed as a browser plugin can navigate and interact with the generated chart. EX1008, 0016; EX1009, 0139. Furthermore, as explained above, the QlikView client can define where the interactive chart is run. The QlikView client can then interact with the chart, on the second website in the absence of the generation interface. As the QlikView Manuals explain, the second website can be hosted on a separate server than the server hosting the first website and the generation interface. *See* EX1009, 0098-99.

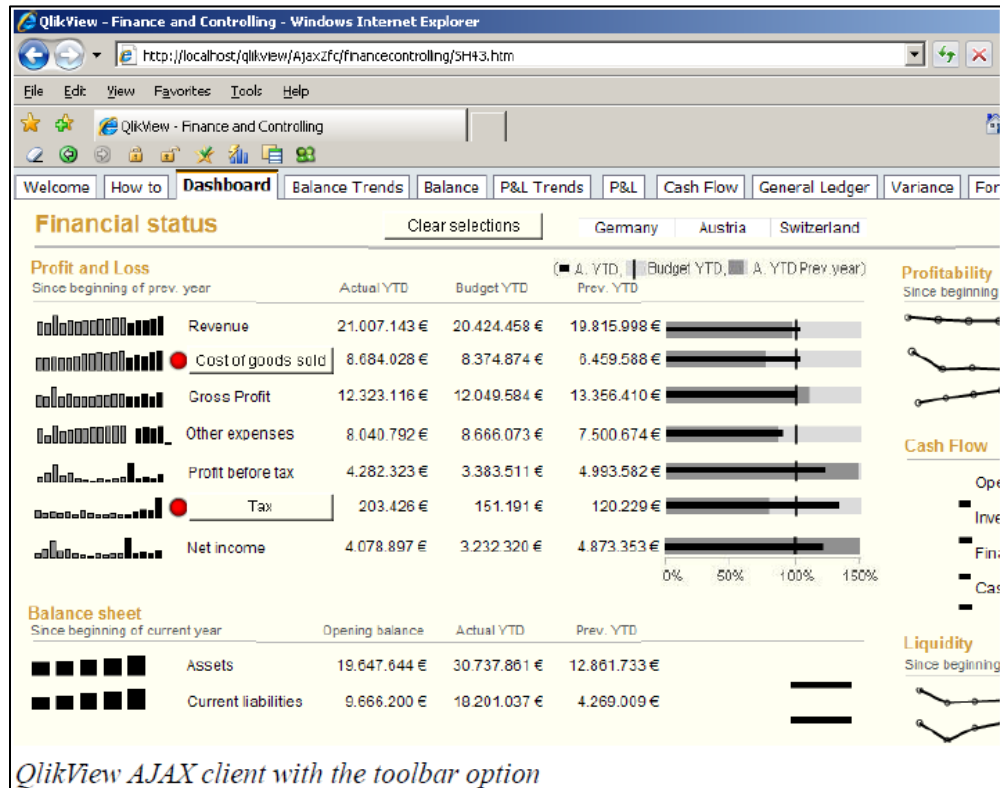
329. When displayed on the second website in the absence of the Chart Wizard's UI ("*the generation interface*"), the interactive chart retains interactivity. For example, when the interactive chart is opened in an applet on an Internet browser outside of QlikView Professional or QlikView Developer, the web client enables a user to render the chart and provide input (e.g., mouse clicks). EX1008, 0013, 0015; EX1009, 0133, 0155-56. For example, the user can make selections by clicking options for the chart via the mouse. EX1008, 0881. Once the second

website with the interactive chart is rendered, the client and chart can also obtain input from the user via “drop-down select in tables,” “slider object[s],” “buttons,” “interactive search feedback,” etc. EX1009, 0016-17, 0019. The webpage further supports right click context menus and fast type change in charts. EX1009, 0018-19. As shown below, data can also be sorted and edited within a chart. EX1009, 0012, 0019, 0133. Additionally, the user can navigate the data using the scrollbars within the charts. EX1009, 0133.



EX1009, 0133.

330. As an additional example, the QlikView AJAX client shown below allows the user to access and interact with charts. EX1009, 0155. The QlikView AJAX client interface is updated asynchronously in response to updates and user interactions, without the need to refresh the page. EX1009, 0155-56.



EX1009, 0155.

331. As a result, the client takes the user's input (e.g., a mouse click) and rerenders the chart using the input, thereby enabling the interactive chart to retain interactivity. EX1009, 0012; EX1008, 0689.

332. For at least these reasons, the QlikView Manuals disclose [1D].

6. [1E] “wherein the selection is made by dragging and dropping the at least one interactive chart from the plurality of charts into a chart box folder connected to the chart box such that retrieve code for data feed displays the at least one interactive chart on the first website”

333. Claim 1 recites “*wherein the selection is made by dragging and dropping the at least one interactive chart from the plurality of charts into a chart box folder connected to the chart box such that retrieve code for data feed displays the at least one interactive chart on the first website.*” The QlikView Manuals in view of Bales disclose this limitation.

334. As discussed for [1A], a user of QlikView selects an interactive chart from the plurality of charts. Once selected, QlikView saves the charts in a “*folder*”:

Target folder for pages: Specifies the folder where the generated pages will be saved. Each page variant (for different client types) will be saved in a separate sub-folder in the specified location.

EX1009, 0134.

335. Each QlikView client has a default chart box folder, listed below:

Client	Directory Folder
Java Object	C:\Program Files\QlikView\Webpages\JavaObjects
QlikX	C:\Program Files\QlikView\Webpages\Plugin
AJAX Zero Footprint	C:\Program Files\QlikView\Webpages\AjaxZFC

EX1009, 0134, 0144, 0157.

336. Although each client has a default chart box folder, the user can choose to save the chart in a separate folder. EX1008, 0807-08. For example, shown below within the “Target folder for pages” box, the user can specify a location to save the generated page. EX1008, 0807-08.

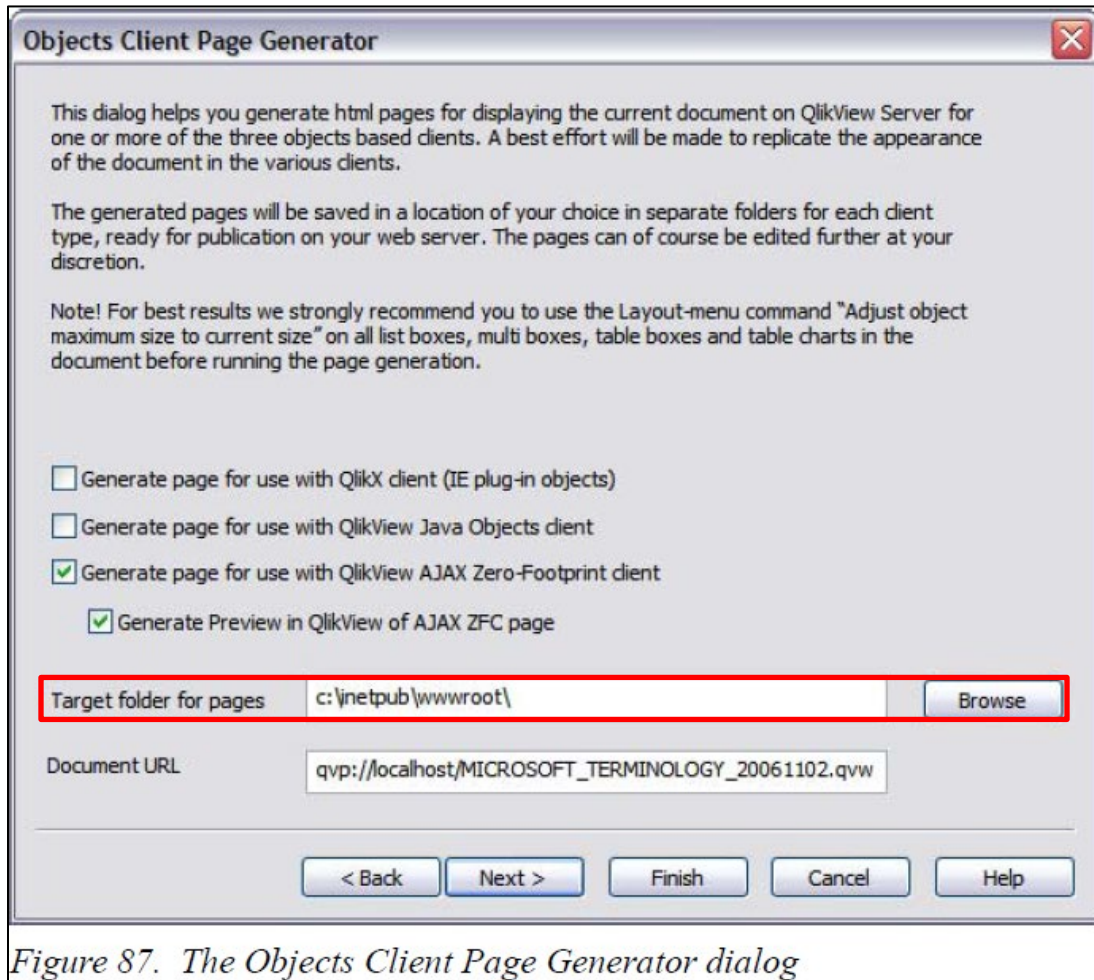


Figure 87. The Objects Client Page Generator dialog

EX1008, 0807 (annotated).

337. Each chart box folder, wherever stored, is “connected to the chart box” because connector code connects the folders (and the QlikView objects contained within) to the chart boxes on the second website. For example, in the

QlikView AJAX client, HTML code includes document connector code. EX1009, 0021. The following code snippets show connector code that connects a chart box folder (and the objects within) to chart boxes.

```
<!--#INCLUDE Virtual="/QvJava/Servers/QlikWebObjects/Qlik-  
Web.asp"-->  
<html>  
<head>  
    <title>.: QlikWeb Objects :.</title>  
</head>  
<body>  
    <% setDocument("qwt_objects.qvw") %>  
    <table border="0" cellspacing="0" cellpadding="0">  
        <tr>  
            <td>  
                <% addOb-  
ject("qwt_objects.qvw", "LB01", 140, 80) %>  
            </td>  
            <td>  
                <% addOb-  
ject("qwt_objects.qvw", "LB02", 250, 100) %>  
            </td>  
        </tr>  
        <tr>  
            <td>  
                <% addOb-  
ject("qwt_objects.qvw", "LB03", 140, 140) %>  
            </td>  
            <td>  
                <% addOb-  
ject("qwt_objects.qvw", "CH01", 250, 150) %>  
            </td>  
        </tr>  
    </table>  
</body>  
</html>
```

Connector Code

EX1009, 0136 (annotated).

338. As noted above in [1B], the “setDocument” command connects the webpage to the document on the QlikView server. EX1009, 0135.

```
<html>
<head>
  <title>QlixTest</title>
  <meta name="vs targetSchema"
  content="http://schemas.microsoft.com/intel-
  lisense/ie5">
  <script type="text/javascript">
  function ClearAll()
  {
    QVPID1.ActiveDocument.ClearAll();
  }
  </script>
</head>
<body>
  <OBJECT id="QVPID1"
  height="1"
  width="1"
  classid="CLSID:74DC8438-E36A-40a0-B750-
  4E2257FA2E41">
    <PARAM
    NAME="SRC"
    VALUE="qvp://extra.qliktech.ideon.se/
    MyQvApp.qvw">
    <PARAM
    NAME="AutoScan"
    VALUE="-1">
  </OBJECT>

  <OBJECT id="Qlix1"
  height="122"
  width="102"
  classid="CLSID:54D93DE9-7DA5-4477-A3DA-
  869ADB41CBBB" >
    <PARAM NAME="ObjectID" VALUE="LB01">
    <PARAM NAME="QvPluginID" VALUE="QVPID1">
  </OBJECT>

  <OBJECT
  id="Qlix2"
  height="288"
  width="448"
  classid="CLSID:54D93DE9-7DA5-4477-A3DA-
  869ADB41CBBB" >
    <PARAM NAME="ObjectID" VALUE="CH01">
    <PARAM NAME="QvPluginID" VALUE="QVPID1">
  </OBJECT>

  <input
  type="button"
  value="ClearAll"
  onclick="ClearAll()"
  id=button2
  name=button2>
</body>
</html>
```

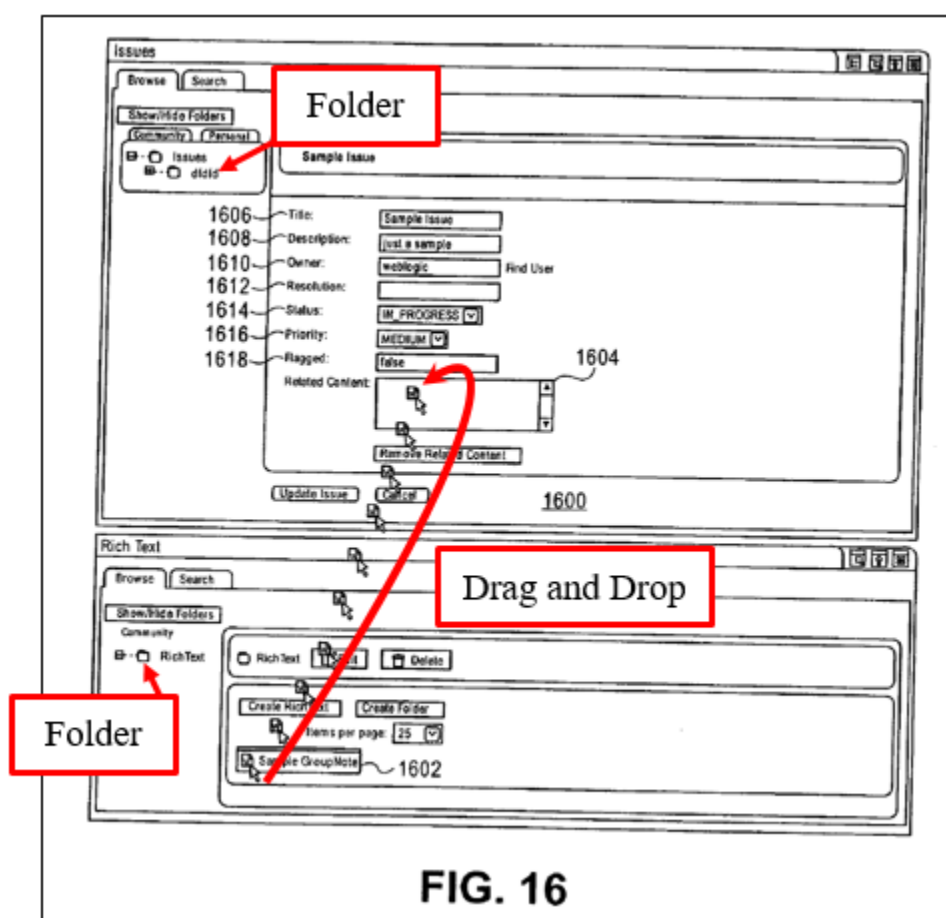
Connector Code

EX1009, 0148-49 (annotated).

The Java applets also connect a chart box folder/objects to chart boxes. *See* EX1009, 0130, 0135.

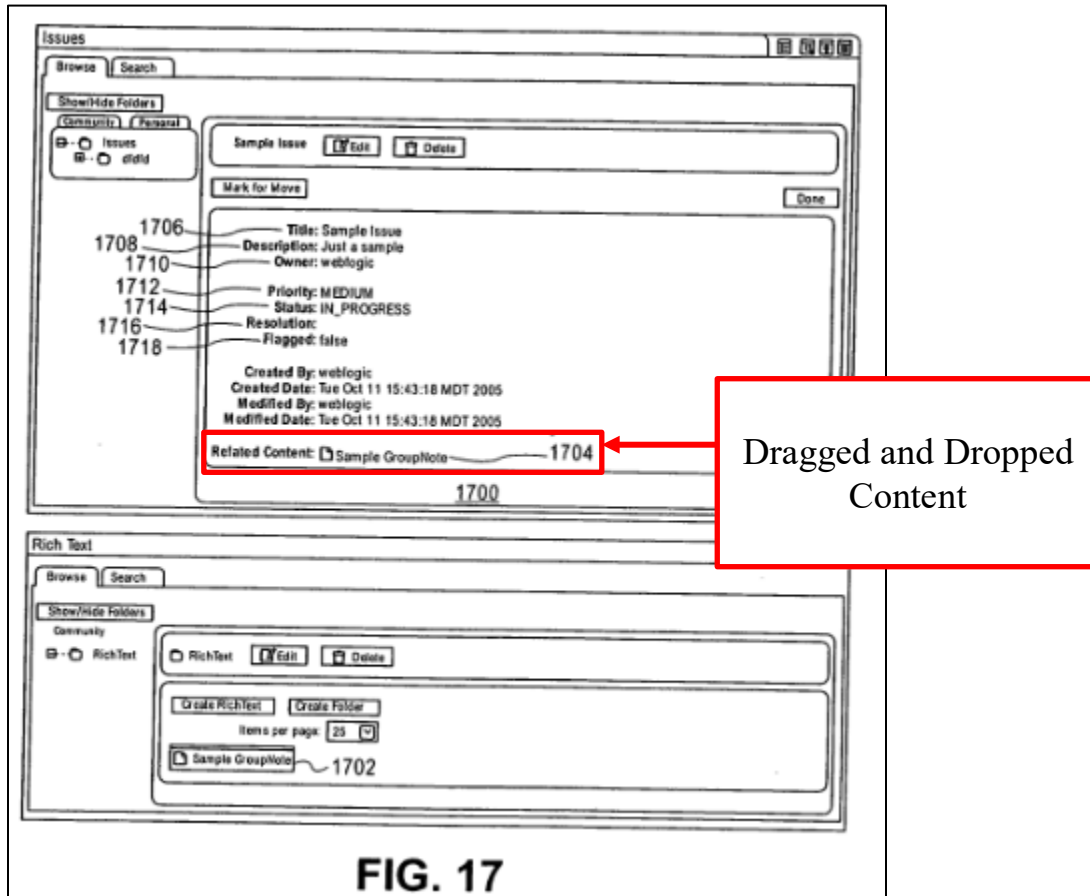
339. The QlikView Manuals therefore disclose “*a chart box folder connected to the chart box.*” Although QlikView users select an interactive chart and save it in “*a chart box folder connected to the chart box,*” the QlikView Manuals do not explicitly disclose making the selection by “*dragging and dropping*” the interactive chart into the chart box folder. Nor do the QlikView Manuals explicitly disclose “*retrieve code for data feed [that] displays the at least one interactive chart on the first website.*” Bales cures these deficiencies.

340. Bales discloses selecting content by “*dragging and dropping*” it from one folder to another folder, as shown below. EX1006, ¶¶158-61. As shown below, the user can drag resource 1602 (“Sample GroupNote”) and drop it onto related content field 1604, creating a relationship between the two resources. EX1006, ¶159. The relationship creates a link between the two resources allowing a user to access one resource from the other. EX1006, ¶¶150-54.



EX1006, FIG. 16 (annotated).

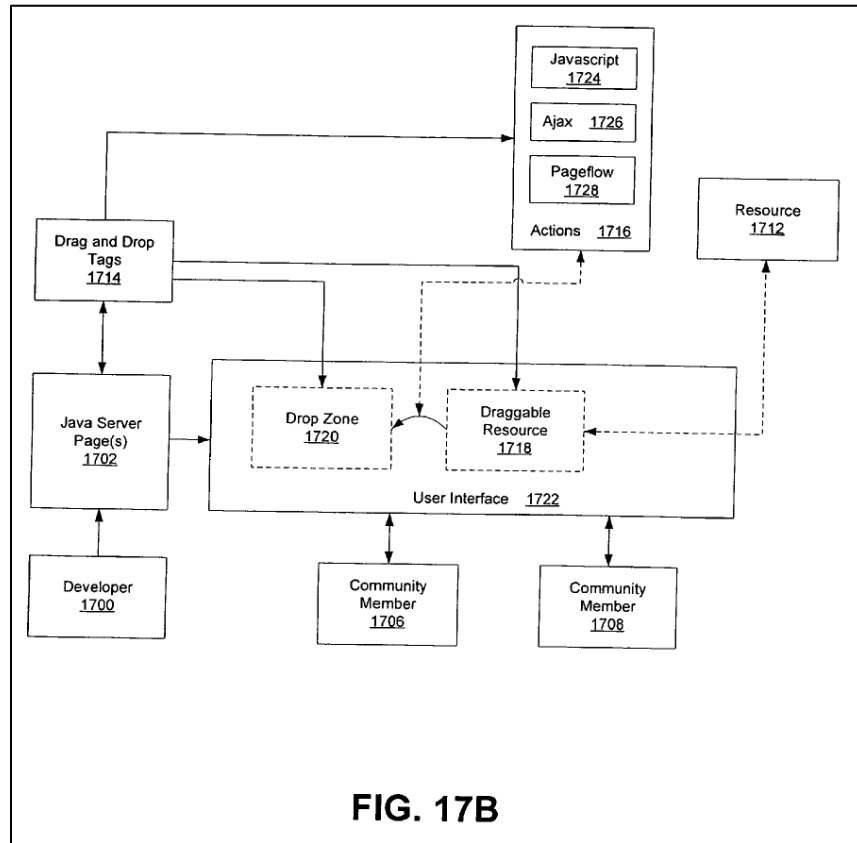
341. Bales further discloses displaying the dropped content. Shown below in Figure 17, the “Sample GroupNote” is subsequently displayed within the “Related Content” field after being dropped. EX1006, ¶160.



EX1006, FIG. 17 (annotated).

342. Bales also discloses dragging-and-dropping selections on Java server pages 1702. EX1006, ¶168. Shown below, a draggable resource 1718 on the user interface 1722 is dropped into drop zone 1720. EX1006, ¶168. The drop zone can be an area of a website. EX1006, ¶168. When combined with the QlikView

Manuals (Section XII.B), the drop zone 1720 would be QlikView's chart box folder, which would still be connected to the chart box via the connector code.



EX1006, FIG. 17B.

343. Additionally, the QlikView Manuals in view of Bales disclose “retrieve code for data feed [that] displays the at least one interactive chart on the first website.” As Bales explains, “[u]pon the user . . . selecting the draggable resource (1718) and dragging and dropping the resource into the resource drop zone, an action (1716) can be performed by the system.” EX1006, ¶168. For example, once dropped, the action can include calling a JavaScript function,

making an AJAX call, executing a page flow action, or calling a form action.

EX1006, ¶¶158, 168.

344. As Bales shows in Figures 16-17 depicted above, the drag and drop action also includes displaying the dragged content on the first website. EX1006, ¶160. For example, as depicted in Figures 16-17, the dragged and dropped content (e.g., “Sample GroupNote”) is displayed in both interfaces. EX1006, FIGs. 16-17. When combined with the QlikView Manuals, the Javascriptfunction, AJAX call, or page flow action would include “*retrieve code for data feed*” and the dragged content displayed on the website would be the interactive chart. Section XII.B.

345. For at least these reasons, the QlikView Manuals in view of Bales disclose [1E].

7. **[1F] “wherein the at least one interactive chart is automatically displayed on the second website using a data feed from the first website to the second website upon the dragging and dropping selection.”**

346. Claim 1 recites “*wherein the at least one interactive chart is automatically displayed on the second website using a data feed from the first website to the second website upon the dragging and dropping selection.*” The QlikView Manuals in view of Bales disclose this limitation.

347. A POSA would have understood that the connector code for QlikView’s clients ([1E], Section XII.C.6) is part of “*a data feed from the first website to the second website*” because the connector code causes the interactive

chart to be displayed in a chart box on the second website. Additionally, as explained for [1B] and [1D] (Sections XII.C.3 and XII.C.5), the charts remain interactive when published. A POSA would have therefore understood the QlikView Manuals to disclose “*wherein the at least one interactive chart is . . . displayed on the second website using a data feed from the first website to the second website.*”

348. The AJAX embodiment discloses this claim element also because it provides asynchronous updates to data shown on the second website. EX1008, 0020-21. The QlikView Manuals state that for AJAX clients, the server sends dynamic updates to charts (e.g., updated data) displayed on the second website:

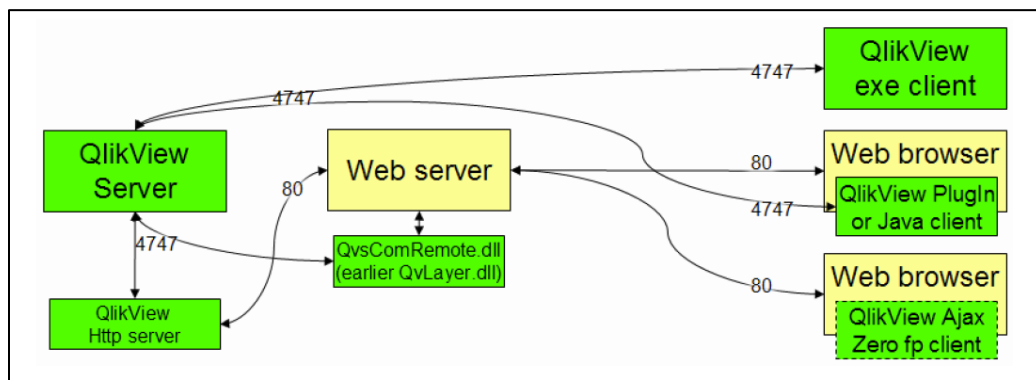
One of the main advantages of the AJAX architecture is the inherent asynchronous update capability to provide quick, incremental updates to the user interface, without requiring a browser page refresh. The QlikView AJAX ZFC provides the environment for the QlikView Server to produce and send Dynamic HTML (DHTML) pages and XML data to the browser running on the client computer and also receive feedback from the user clicking in those pages.

EX1009, 0155-56; *see also* EX1008, 0020-21; EX1009, 0012, 0014 (AJAX updates).

349. As a result, the AJAX client UI is updated without requiring the page to be refreshed. EX1009, 0155-56. In other words, QlikView “**automatically** generate[s] AJAX ZFC pages” EX1009, 0016. As a POSA would have

understood, automatically displaying updates using a data feed from the developer website to the client website and vice versa was a core feature of AJAX.

350. The QlikView Manuals show data feeds that display the interactive charts upon selection of the chart for publication. For example, the QlikView Server communicates with web clients via port 4747, and the HTTP Server communicates with AJAX clients via port 80. EX1009, 0098.



EX1009, 0098.

351. For these additional reasons, the QlikView Manuals disclose “*wherein the at least one interactive chart is . . . displayed on the second website using a data feed from the first website to the second website.*” However, as discussed in Section XII.B, the QlikView Manuals do not explicitly disclose an automatic display “*upon [a] dragging and dropping selection.*” Bales cures this deficiency.

352. As discussed for [1E], Bales discloses a drag-and-drop selection. EX1006, ¶¶158-68, FIGs. 16-17B. Upon making the drag-and-drop selection in Bales, the content is “*automatically displayed on the second website using a data*

feed from the first website to the second website.” As Bales explains, “when a [user] drops the draggable resource onto a drop zone, a dynamic content call can be made to a remote server, for updating the drop zone area of the user interface with an HTML fragment.” EX1006, ¶168. The dynamic content call can be “an AJAX call.” EX1006, ¶168. This AJAX call uses a data feed to automatically display updates on a second website. EX1006, ¶¶139, 147.

353. With AJAX, a data feed from a first website to a second website automatically displays updates upon the drag-and-drop selection. For example, Bales uses AJAX to update stock market data within a dynamic portion on a second website. EX1006, ¶¶122, 125, 146. Using AJAX, only the portion of the page showing the stock market data is updated, alleviating the need to reload the entire web page. EX1006, ¶125. At the time Bales was published (2007), stock prices were commonly displayed in line graphs (charts) that were updated throughout the day. Accordingly, a POSA would have understood that dropping an updated stock-price chart in Bales’s drop zone would automatically display the updated chart on the second website using a data feed from the first website to the second website. EX1006, ¶¶125, 128, 139-40, 146-47, 158, 168. Bales makes clear that it uses a data feed to display updated content on the second website “automatically”:

In step 1412, the server can then return the relevant information back to the original pinging member_A. For example, the notifications manager can **update the user interface portion of member_A's desktop automatically to display the updated information**, without requiring any input from member_A. The notifications manager can be AJAX based.

EX1006, ¶147.

354. Bales further discloses:

[A] **value** within a portlet **may be dynamically updated** each time the notification manager updates it, without re-posting the entire browser. Thus, a granular and preemptive updating process is achieved, yet **requiring very little or no user interaction**.

EX1006, ¶147.

355. For at least these reasons, the QlikView Manuals in view of Bales disclose [1F].

D. Dependent Claim 3

356. Claim 3 depends from claim 1 and recites “*displaying one or more of the plurality of interactive charts in the chart box on the second website.*” The QlikView Manuals disclose this limitation for the same reasons they disclose [1B]. See Section XII.C.3.

357. For example, QlikView displays the chart embedded in a chart box on a second website, as shown below. As The QlikView Manuals explain, “[t]he

QlikView Objects Clients (Java Objects, QlikX, and AJAX) are based on the concept of placing individual sheet objects from a QlikView document in an HTML environment.” EX1009, 0009.



QlikView Java Objects client in Internet Explorer.

EX1009, 0133 (annotated).

358. For these reasons and the reasons provided for claim 1, the QlikView Manuals in view of Bales render obvious claim 3.

E. Dependent Claim 4

359. Claim 4 depends from claim 1 and recites “*displaying a plurality of interactive charts embedded in a plurality of chart boxes on the second website.*”

The QlikView Manuals in view of Bales disclose this limitation.

360. As discussed for [1B], QlikView displays an interactive chart embedded in a chart box on a second website. And QlikView is not limited to a single interactive chart embedded in a single chart box. For example, using the “addObject” function, the user can insert an interactive chart within a chart box. EX1009, 0136. As the script shows below, multiple QlikView objects (e.g., charts) are displayed in multiple chart boxes. In this example, the code defines four chart boxes (two rows <tr>, each row having two cells <td>). EX1009, 0136. A different QlikView object from the QlikView file (qwt_objects.qvw) is embedded in each chart box.

```
<!--#INCLUDE Virtual="/QvJava/Servers/QlikWebObjects/Qlik-  
Web.asp"-->  
<html>  
<head>  
  <title>.: QlikWeb Objects :.</title>  
</head>  
<body>  
  <% setDocument("qwt_objects.qvw") %>  
  <table border="0" cellspacing="0" cellpadding="0">  
    <tr>  
      <td>  
        <% addObject-  
          ject("qwt_objects.qvw", "LB01", 140, 80) %>  
      </td>  
      <td>  
        <% addObject-  
          ject("qwt_objects.qvw", "LB02", 250, 100) %>  
      </td>  
    </tr>  
    <tr>  
      <td>  
        <% addObject-  
          ject("qwt_objects.qvw", "LB03", 140, 140) %>  
      </td>  
      <td>  
        <% addObject-  
          ject("qwt_objects.qvw", "CH01", 250, 150) %>  
      </td>  
    </tr>  
  </table>  
</body>  
</html>
```

EX1009, 0136; *see also* EX1009, 0147-49 (QlikX displaying multiple objects).

361. As a POSA would have understood, the other web clients are also not limited to displaying a single interactive chart embedded in a single chart box.

Rather, the clients can display multiple objects (charts) in a QlikView document using multiple chart boxes. EX1009, 0129-166.

362. For these reasons and the reasons provided for claim 1, the QlikView Manuals in view of Bales render obvious claim 4.

XIII. No Objective Indicia of Nonobviousness

363. In reaching my opinions in this declaration, I considered whether there were any objective indicia of nonobviousness. The basis for my understanding of objective indicia of nonobviousness is in Section IV.B.

364. I understand that any objective indicia of nonobviousness must have a nexus, or a close connection, to the novel aspects of the claims. I understand that objective indicia of nonobviousness include the following: commercial success, long-felt but unmet need, failure of others, copying, praise in the industry, and unexpectedly superior results as compared with the closest prior art.

365. Considering the information I have reviewed, I am not aware of any objective indicia that might support the patentability of the claims of the '000 patent. Further, from my review of the file history (EX1002) of the '000 patent, I did not see iCharts raising any objective indicia of nonobviousness during prosecution. I am not otherwise aware of any publicly available evidence of objective indicia of nonobviousness.

366. Therefore, it is my opinion that no objective indicia of nonobviousness support the patentability of the claims of the '000 patent.

XIV. Conclusion

In signing this declaration, I recognize that the declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case and that cross-examination will take place within the United States. If cross-examination is required of me, I will appear for cross-examination within the United States during the allotted time.

I declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code.

Respectfully submitted,

A handwritten signature in black ink, appearing to read 'Chris Schmandt', written over a horizontal line.

Christopher Schmandt

Date: 26 September 2024