

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

TABLEAU SOFTWARE, LLC
Petitioner

v.

ICHARTS LLC
Patent Owner

Case IPR2024-01388
Patent No. 9,712,595

**DECLARATION OF CHRISTOPHER SCHMANDT, IN SUPPORT OF
PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 9,712,595**

Mail Stop “PATENT BOARD”

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

Tableau EX1003
IPR2024-01388
U.S. Patent No. 9,712,595

TABLE OF CONTENTS

I.	Introduction	1
II.	List of Documents Considered	3
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. 9,712,595.....	16
A.	Claims.....	25
B.	Prosecution History of the '595 Patent.....	27
C.	Provisional Application No. 61/077,728	28
D.	New Matter in U.S. Application No. 12/372,696	32
E.	Level of Ordinary Skill in the Art	35
VI.	Technology Overview.....	35
A.	Data Visualization Tools.....	35
B.	Flash	52
C.	JavaScript	54
D.	Drag and Drop	56
E.	Panes	59
VII.	Overview of the Prior Art	63

A.	Rostoker discloses a platform for generating, modifying, configuring, or adapting interactive charts.	63
B.	Clark discloses a system for publishing charts to an enterprise website.....	68
C.	Jou discloses a visualization system for generating interactive charts.	70
D.	Couckuyt discloses a tool with a chart pane and data pane used to design interactive charts with drag-and-drop.....	73
E.	The QlikView Manuals disclose a visual analytics platform to create and publish interactive charts.....	78
VIII.	Grounds of Unpatentability.....	86
IX.	Ground 1: Rostoker renders obvious claims 1, 2, 4-8, 10-14, and 16-18.....	87
A.	Independent Claim 1	87
B.	Independent Claims 7 and 13	127
C.	Claims 2, 8, and 14	130
D.	Claims 4, 10, and 16	132
E.	Claims 5, 11, and 17	133
F.	Claims 6, 12, and 18	135
X.	Ground 2: Rostoker in view of Clark renders obvious claims 1, 2, 4-8, 10-14, and 16-18.....	137
XI.	Grounds 3 and 4: Rostoker, alone or in view of Clark, and further in view of Jou renders obvious claims 1, 2, 4-8, 10-14, and 16-18.	142
XII.	Grounds 5-8: Rostoker, alone or in view of Clark and/or Jou, and further in view of Couckuyt renders obvious claims 3, 9, and 15.....	146
A.	Claim 3	146
B.	Claim 9	153

C.	Claim 15	154
XIII.	Ground 9: The QlikView Manuals render obvious claims 1, 2, 4-8, 10-14, and 16-18.	154
A.	A POSA would have been motivated to combine the QlikView Manuals.	155
B.	Independent Claim 1	156
C.	Independent Claims 7 and 13	200
D.	Claims 2, 8, and 14	205
E.	Claims 4, 10, and 16	206
F.	Claims 5, 11, and 17	209
G.	Claims 6, 12, and 18	211
XIV.	Ground 10: The QlikView Manuals in view of Couckuyt renders obvious claims 3, 9, and 15.	214
A.	A POSA would have been motivated to combine the teachings in the QlikView Manuals and Couckuyt.	215
B.	Claim 3	218
C.	Claim 9	222
D.	Claim 15	222
XV.	No Objective Indicia of Nonobviousness	223
XVI.	Conclusion.....	225

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. I understand that this proceeding involves U.S. Patent No. 9,712,595 (EX1001, “the ’595 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 ’595 patent is currently assigned to iCharts LLC. I also understand that the ’595 patent names two inventors: Seymour Duncker and Tyron Montgomery. EX1001, (75).

3. I have reviewed and am familiar with the specification and the claims of the ’595 patent. The ’595 patent was filed on October 31, 2014 and claims priority as a continuation of U.S. Patent Application No. 13/962,236, which was filed on August 8, 2013 and now U.S. Patent No. 9,270,728, which is a continuation of U.S. Patent Application No. 12/372,696, which was filed on February 17, 2009 and now U.S. Patent No. 8,520,000, which is a continuation-in-part of U.S. 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 '595 patent also claims priority to Provisional Application No. 61/077,728, which was filed on July 2, 2008. EX1001, (60).

4. I understand the '595 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 '595 patent at column 1, lines 1-10.

5. I have reviewed and am familiar with the file history of the '595 patent. The file history of the '595 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 '595 patent:

- EX1005: U.S. Publication No. 2008/0127052 to Rostoker;
- EX1006: U.S. Patent No. 6,906,717 to Couckuyt *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); and
- EX1010: U.S. Publication No. 2006/0112123 to Clark *et al.*

7. I have been asked to provide my technical review, analysis, insights, and opinions about the '595 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. 9,712,595 to Duncker <i>et al.</i>
1002	U.S. Patent Prosecution History of U.S. Patent No. 9,712,595
1005	U.S. Publication No. 2008/0127052 to Rostoker (“Rostoker”)
1006	U.S. Patent No. 6,906,717 to Couckuyt <i>et al.</i> (“Couckuyt”)
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 B)
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	Nikolaidou <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)
1066	Microsoft Computer Dictionary 5th Edition (2002)
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.

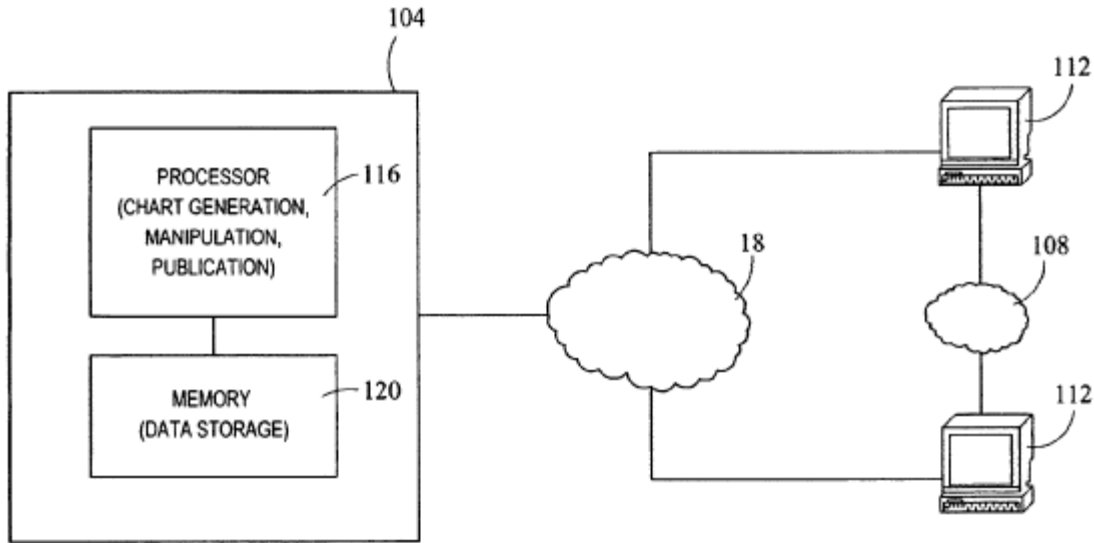
32. My opinion below explains how a POSA would have understood the technology described in the references I have identified herein around the February 17, 2009 timeframe. However, my opinions herein would not be affected if the '595 patent is found to be entitled to an earlier priority date.

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. 9,712,595

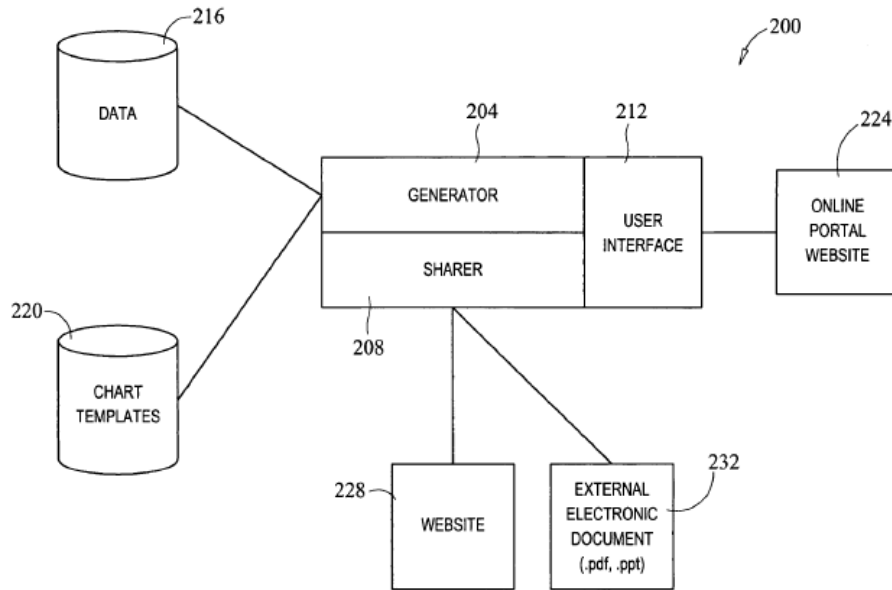
34. The '595 patent describes “[s]ystems and methods for generating and sharing interactive charts.” EX1001, Abstract.

35. The '595 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, 6:61-63. “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, 7:28-31.



EX1001, FIG. 1.

36. Figure 2 (below) shows “interactive chart online portal 200” EX1001, 7:59-60. “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, 7:65-8:1. “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, 8:7-9. “Users upload or enter data at the website 224 through the user interface 212 which stores the data in the data store 216.” EX1001, 8:14-16. “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, 8:16-18.



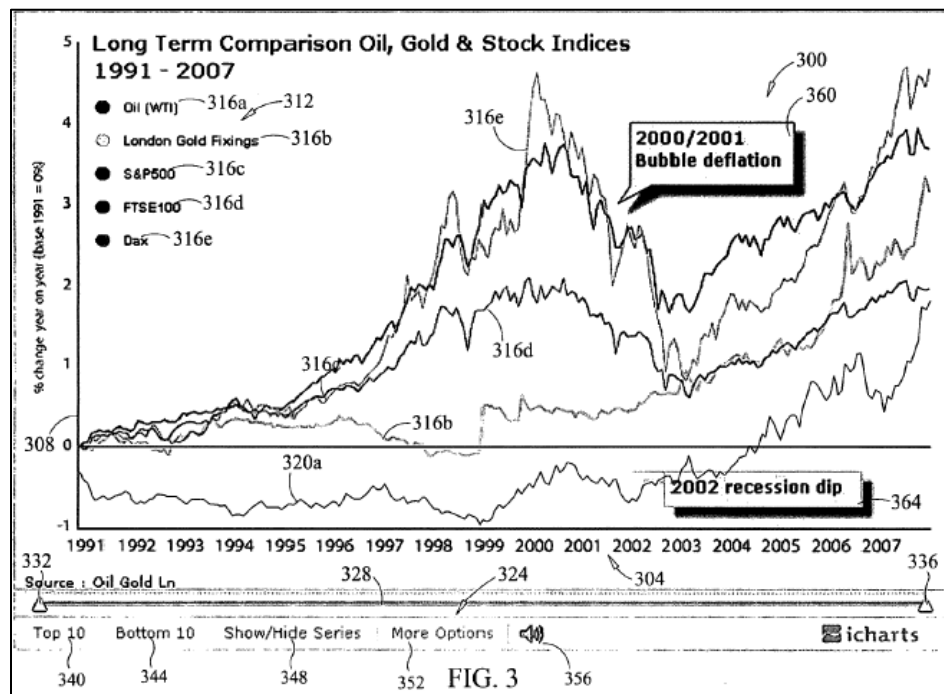
EX1001, FIG. 2.

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, 8:20-22. “The generated interactive chart can also be embedded in an external document 232 as an interactive chart or a static chart.” EX1001, 8:22-24. The ’595 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, 1:31-34, 8:25-27, 13:53-55.

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, 8:60-61. 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, 8:36-40.

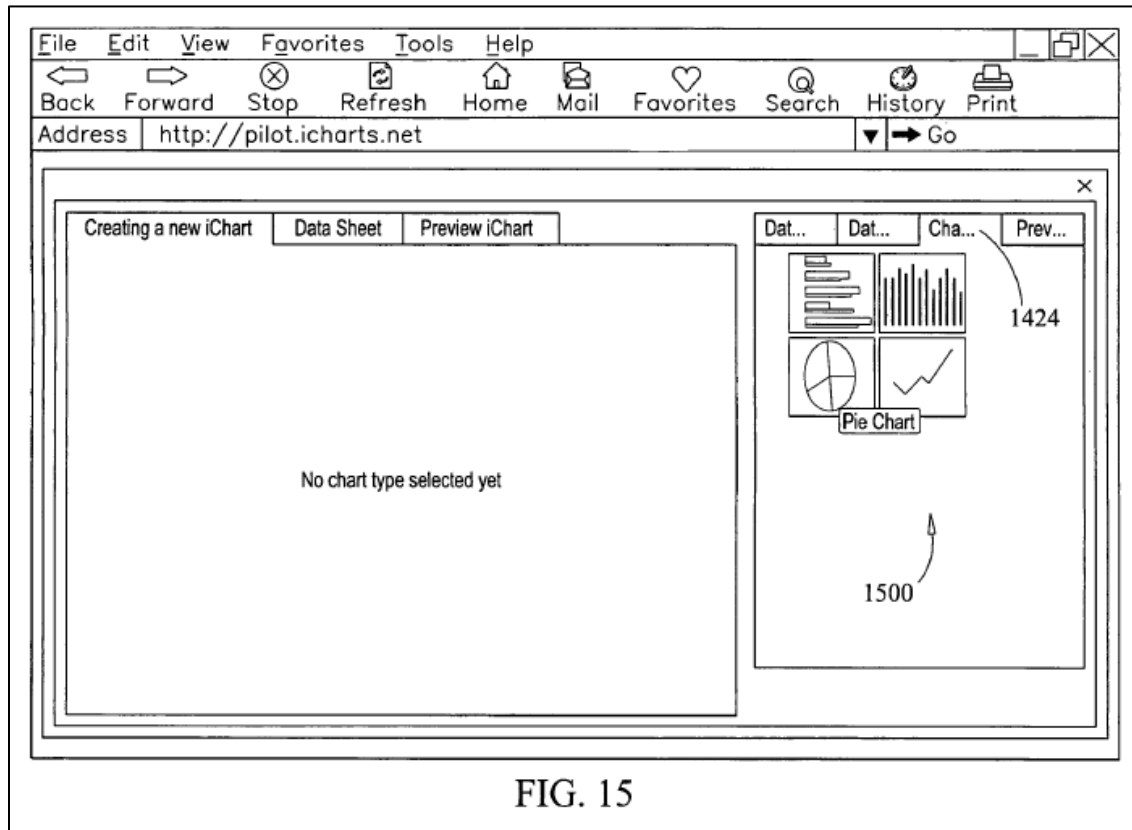
39. The '595 patent refers to “interactive” charts broadly to include different interactive features. For example, shown below, “chart 300 has interactive features that allow a user to interact with the chart to, for example, modify or change their view of the chart, learn more about the chart data, and the like.” EX1001, 9:25-28. “Exemplary interactive features include a zoom feature, a data mouse-over popup window, a data hide/share feature, an audio playback feature, a drill down feature, a data gliding feature and the like.” EX1001, 9:28-31.



EX1001, FIG. 3.

40. The '595 patent also describes an embodiment that provides an “online portal or user interface” that “allows users to drag and drop the data from a

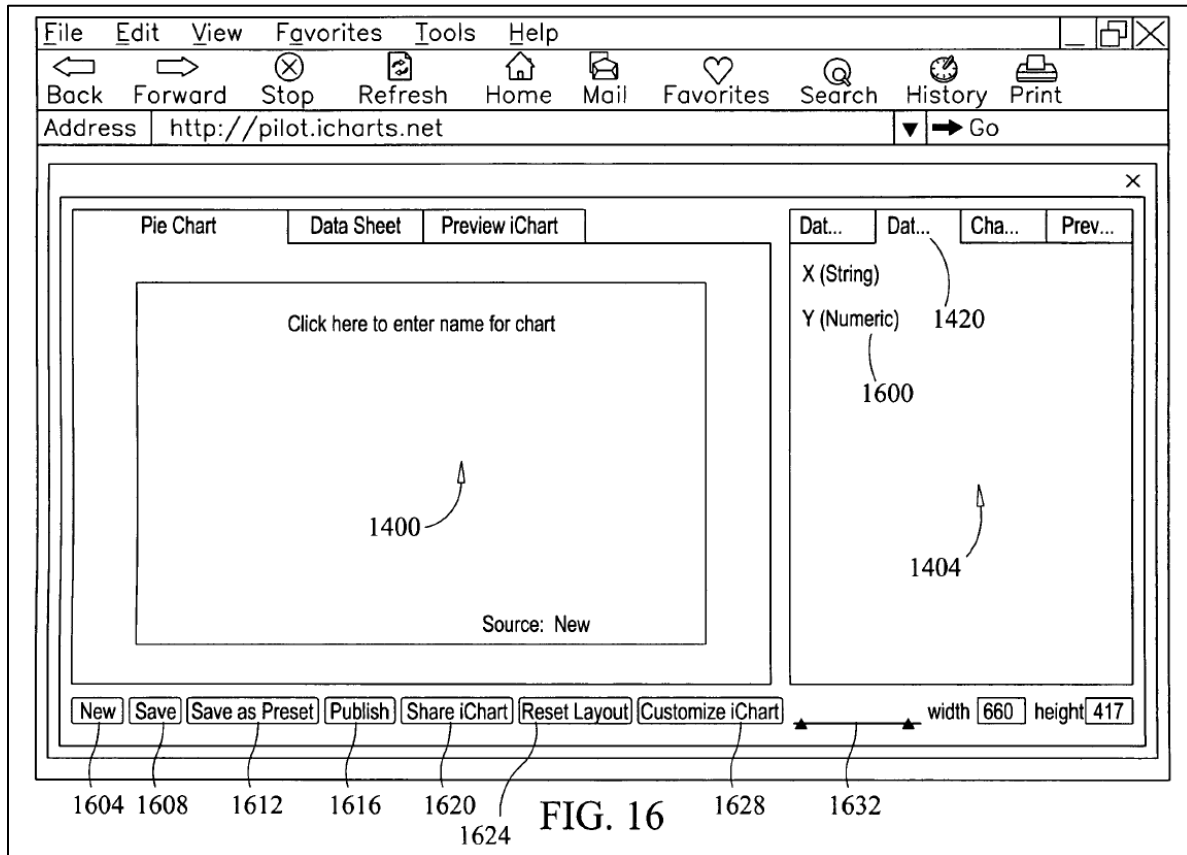
data pane to a charting pane, automatically creating the interactive chart.” EX1001, 5:23-26. In one embodiment shown below, “[w]hen a user clicks on the charting pane 1400, several chart types 1500 are displayed in the chart type tab 1424 of the data pane 1404, as shown in FIG. 15.” EX1001, 11:49-51.



EX1001, FIG. 15.

41. “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, 11:58-60. “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, 11:60-62. “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, 11:62-67.



EX1001, FIG. 16.

42. “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, 12:8-10. “The user can then manipulate the chart 1700 to have the look and feel and interactive features that the user desires” EX1001, 12:23-25.

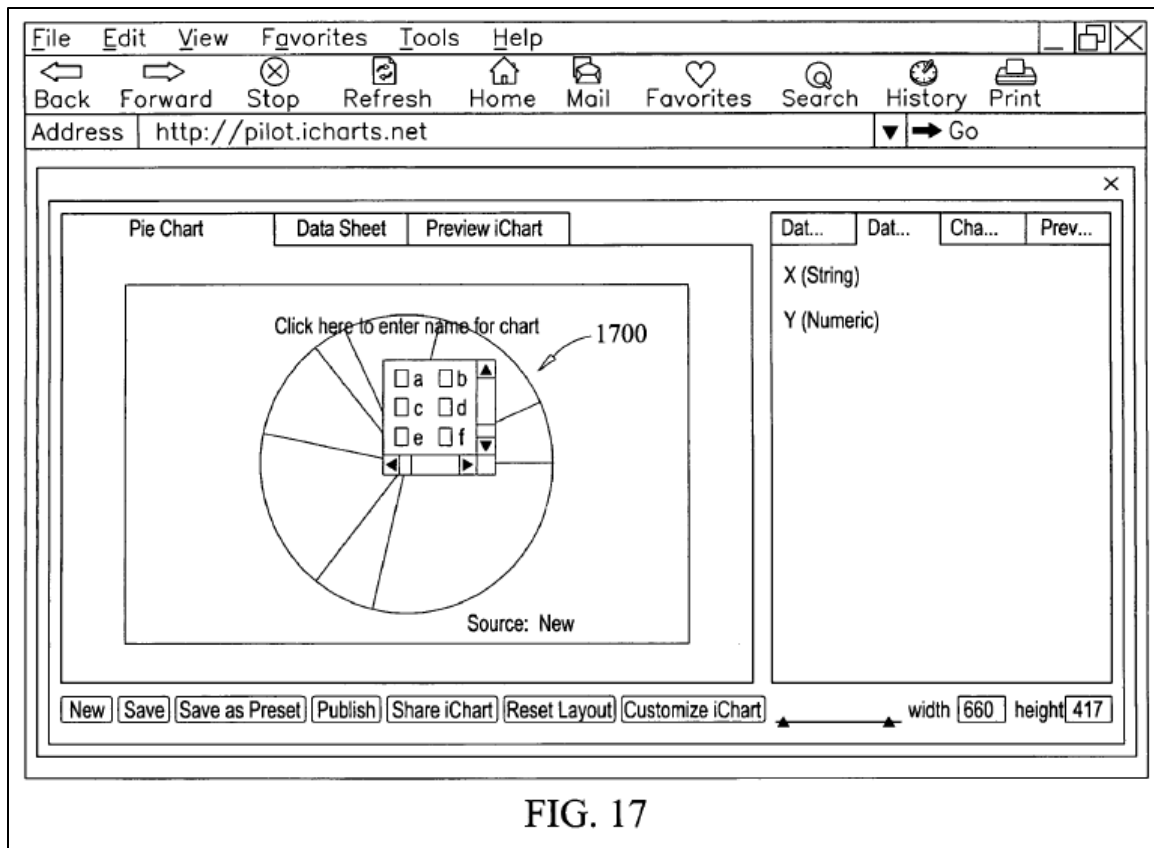
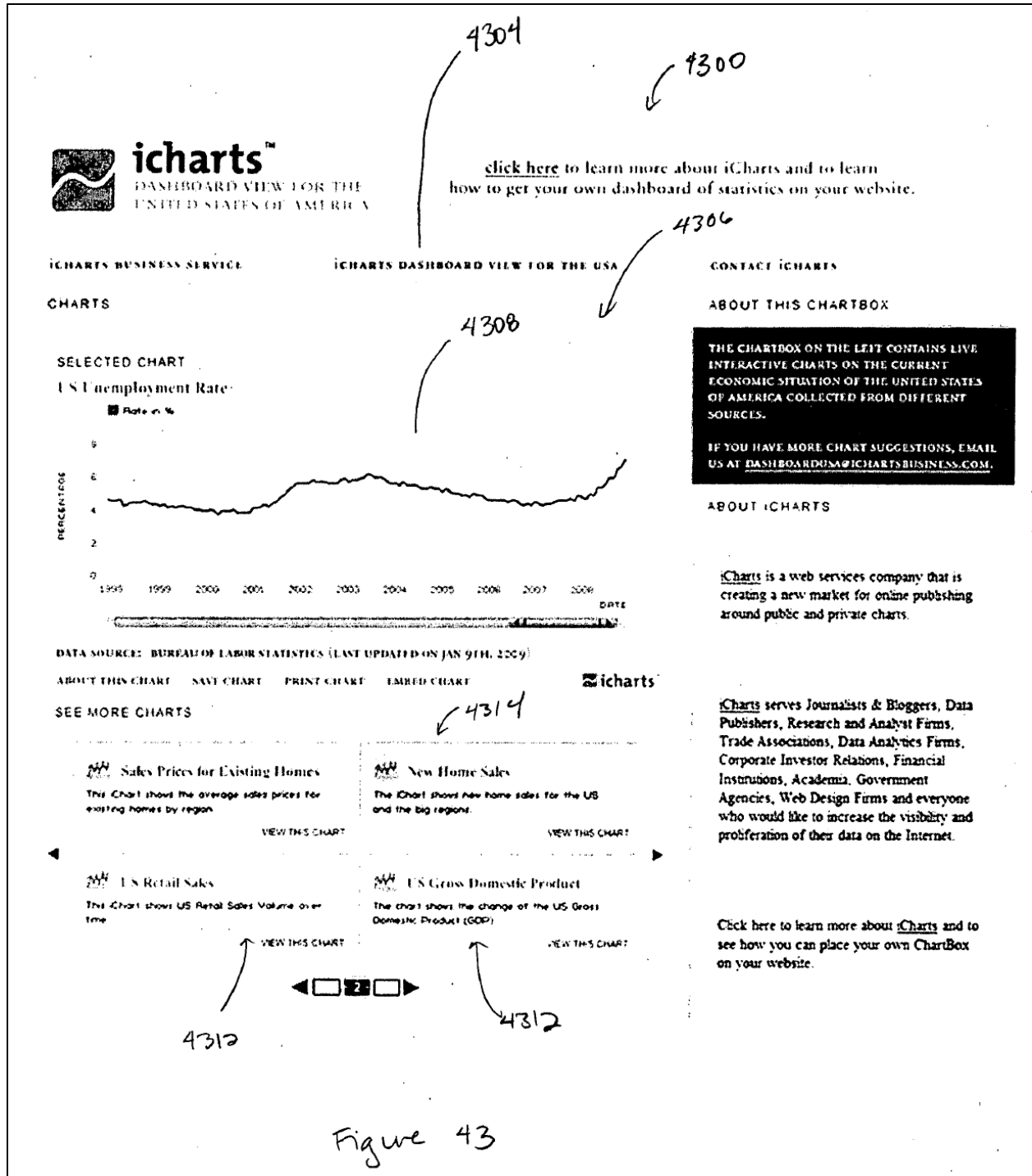


FIG. 17

EX1001, FIG. 17.

43. The '595 patent further 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, 2:14-26.

44. Figure 43 below “illustrates an exemplary screen shot of a business services application website 4300.” EX1001, 16:4-6. “The [] 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, 16:6-10. “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, 16:10-13.



EX1001, FIG. 43.

A. Claims

45. The '595 patent issued with 18 claims, three of which are independent claims. Independent claims 1, 7, and 13 are shown below. As I show below, the claims recite a system/method/computer readable storage medium that receives “*chart data*” and “*a selection of a chart template*,” “*generate[s] an interactive chart*” via “*a user interface on a first website*,” and “*transmits the interactive chart to a second website for display*.”

Independent Claim 1

A system for creating interactive charts, the system comprising:

memory that stores a chart template;

a processor, wherein the processor executes instructions stored in the memory, causing the processor to:

generate a user interface on a first website,

receive chart data via the user interface,

receive a selection of the chart template via the user interface,

and

generate an interactive chart including the chart data and the chart template; and

a network interface that, in response to generation of the interactive chart, transmits the interactive chart to a second website for display without the execution of the instructions

stored in the memory to generate the user interface of the first website, wherein the second website receives an input from a second user and updates the interactive chart after the second website receives the input.

EX1001, 18:40-59.

Independent Claim 7

A method for creating interactive charts, the method comprising:

generating a user interface on a first website;

receiving a selection of a chart template via the user interface;

receiving entry of chart data via the user interface;

generating an interactive chart including the chart data and the chart template;

transmitting the interactive chart to a second website for display in response to generation of the interactive chart, wherein:

the interactive chart is displayed via the second website without generating the user interface of the first website,

the second website receives an input from a second user, and

the interactive chart is updated after the second website receives the input.

EX1001, 19:11-28.

Independent Claim 13

A non-transitory computer readable storage medium having embodied thereon a program executable by a processor for creating interactive charts, the method comprising:

generating a user interface on a first website;

receiving a selection of a chart template via the user interface;

receiving entry of chart data via the user interface;

generating an interactive chart including the chart data and the chart template;

transmitting the interactive chart to a second website for display in response to generation of the interactive chart, wherein:

the interactive chart is displayed via the second website without generating the user interface of the first website,

the second website receives an input from a second user, and

the interactive chart is updated after the second website receives the input.

EX1001, 20:3-22.

B. Prosecution History of the '595 Patent

46. The '595 patent issued from U.S. Patent Application No. 14/530,309 ("the '309 application"), which was filed on October 31, 2014. EX1001, (21), (22).

47. On April 24, 2017, the Examiner initiated a telephonic interview. During the interview, the Examiner proposed allowable subject matter to iCharts' representative and proposed Examiner's amendments. EX1002, 0182.

48. On May 4, 2017, the Office issued a Notice of Allowance after iCharts accepted the Examiner's amendments. EX1002, 0166.

C. Provisional Application No. 61/077,728

49. The '595 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.

50. 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 '595 patent. In other words, I understand that the '728 provisional application must describe the claimed invention in the '595 patent in sufficient detail that one skilled in the art can reasonably conclude that the inventor had possession of the claimed invention.

51. 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 '595 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.

52. The specification in the '728 provisional application was very short (13 paragraphs, a little over one page) and different than the '595 patent. *See* EX1013, 0045-46. The '728 provisional application includes some figures that resemble figures in the '595 patent. *See* EX1013, 0054-73. 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 '595 patent are not in the '728 provisional application).

53. There are key omissions from the '728 provisional application that were later claimed in the '595 patent. For example, the '728 provisional application does not describe “*a network interface that, in response to generation of the interactive chart, transmits the interactive chart to a second website for display without the execution of the instructions stored in the memory to generate the user interface of the first website*” (claim 1) or “*transmitting the interactive chart to a second website for display in response to generation of the interactive chart*” (claims 7 and 13). Rather, the inventors added these concepts to the alleged invention *after* the '728 provisional application.

54. iCharts introduced the concept of transmitting the interactive chart via “*a network interface*” to “*a second website*” and displaying the interactive chart on

“[the] second website for display without the execution of the instructions stored in the memory to generate the user interface of the first website” in Figures 43-46 and paragraphs [0140] – [0149] of the filed specification on February 17, 2009.

EX1002, 0033-36 (specification pages 27-30), 0093-96. The portions of the specification that describe the feature are reproduced below.

[0140] FIG. 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 FIG. 44. Referring to FIGS. 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.** 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).**

EX1002, 0033 (specification page 27).¹

[0149] The software 4626 may further be transmitted or received over a network 4628 via the network interface device 4622.

EX1002, 0036 (specification page 30).

55. This description of generating an interactive chart by a first website, transmitting the interactive chart to a second website by a network interface, and

¹ Emphasis added unless otherwise indicated.

displaying the interactive chart on the second website were not in the '728 provisional application. The '728 provisional application states merely that a user can “publish the chart online” and that “the service is created using a flex engine (e.g., Adobe’s flash flex development platform).” EX1013, 0045. Again, this concept was later added to the '595 patent and its figures.

56. The claims of the '595 patent include additional elements that are not in the '728 provisional application. The '728 provisional application does not describe the “*data pane*” as recited in claims 3, 9, and 15 of the '595 patent. *See* EX1013, 0045 (referring to a “data sheet tab” instead of a “data pane”). Again, this concept was later added to the '595 patent and its figures.

57. Nor does the '728 provisional application describe a “*computer system*,” or “*a non-transitory computer readable storage medium*” as recited in the '595 patent. Rather, iCharts added the computer system when submitting new figures for the '595 patent (e.g., Figures 1, 2, 46). *See* EX1002, 0043-44, 0096. iCharts also added new features for the interactive charts, including “*an audio feature, a video feature, an animation feature, and a drill-down feature, [and/or] a data gliding feature*” as recited in claims 5, 11, and 17 of the '595 patent and encompassed in claims 4, 10, and 16 of the '595 patent.

58. As these examples show, the '728 provisional application does not provide sufficient written description of any of the '595 patent claims.

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

59. The '595 patent is related to U.S. Application No. 12/372,696 (filed on February 17, 2009 and issued as U.S. Patent No. 8,520,000) (“the '696 application”), which 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), 1:7-17. 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.

60. I understand that there is simply no reason to presume that claims in a CIP application are entitled to the effective filing date of an earlier-filed 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 '595 patent. In other words, I understand that the '802 application must describe the claimed invention in the '595 patent in sufficient detail that one skilled in the art can reasonably conclude that the inventor had possession of the claimed invention.

61. 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 '595 patent as of September 5, 2008. In other words, the '802 application lacks a complete concept of the later-claimed invention.

62. iCharts added new concepts after the '802 application's filing date. For example, the '309 application included three figures (i.e., Figures 43-45), which are not in the '802 application. EX1002, 0093-95.

63. This new matter is encompassed in the claims of the '595 patent. For example, claim 1 recites "*a network interface that, in response to generation of the interactive chart, transmits the interactive chart to a second website for display without the execution of the instructions stored in the memory to generate the user interface of the first website.*" Claims 7 and 13 both recite "*transmitting the interactive chart to a second website for display in response to generation of the interactive chart.*" As a POSA would have understood, the embodiment depicted in Figures 43-45 is new and encompassed within these claim limitations.

64. For example, the embodiment depicted in Figures 43-45 allows a user to "generate charts" and "automatically update the charts displayed in [a] chart box without reprogramming the website for their blog using another website." EX1001, 16:10-17. "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." EX1001, 16:64-67. For this display to occur, the new matter refers to "retrieve code to fetch the interactive chart(s) . . . EX1001, 16:26-29. "Alternatively, the embed code may include a data feed, such as an ATOM feed, RSS feed, XML feed, etc., that is used to automatically update the interactive

chart(s) displayed in the chart box 4304.” EX1001, 16:29-32. As a POSA would have understood, such configuration “*transmits the interactive chart to a second website for display.*”

65. The ’309 application included additional new matter that relates to transmitting the interactive chart to a second website for display in response to generation of the interactive chart. For example, the ’309 application added the following new matter to its specification:

An online portal may also be provided to allow users to **automatically embed interactive chart(s) in another website** without reprogramming the code of the website each time the interactive chart(s) are changed or new interactive chart(s) are added.

EX1034, Abstract (emphasis added).

The interactive chart may be **automatically displayed using a data feed from the first website to the second website.**

EX1034, 10 (emphasis added).

The interactive charts can also be **embedded in other websites without the need to reprogram the other websites** each time an interactive chart is embedded in those websites using a chart box tool.

EX1034, 19 (emphasis added).

66. Because this new matter is encompassed within the ’595 patent claims but not disclosed in the ’802 application, a POSA would not have reasonably

concluded that the inventors had possession of the full scope of any claim in the '595 patent as of the filing date of the '802 application.

E. Level of Ordinary Skill in the Art

67. A person of ordinary skill in the art (“POSA”) in the technical field of the '595 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.

68. 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.

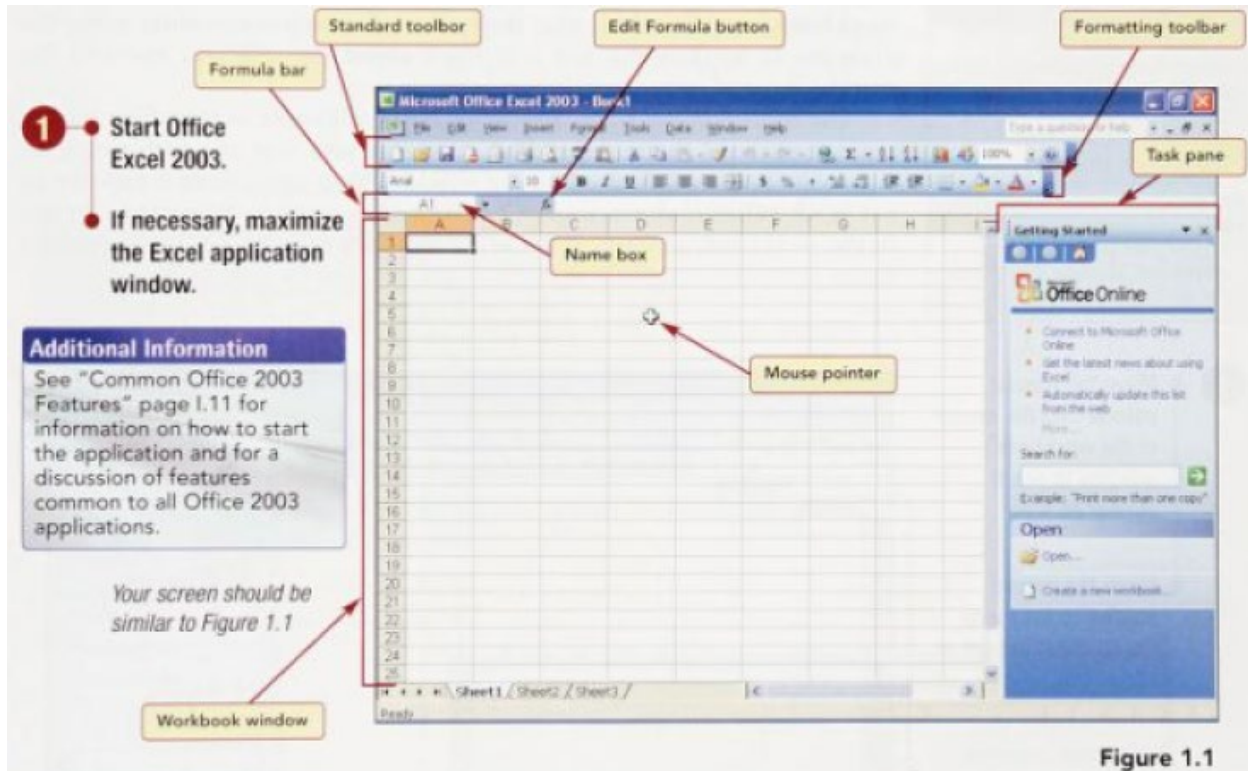
VI. Technology Overview

69. As I explain in the following sections, data visualization tools existed long before the '595 patent, as did interactive charts, data/chart panes, and drag-and-drop techniques.

A. Data Visualization Tools

70. The '595 patent itself recognizes that data visualization tools such as Microsoft Excel existed before the alleged invention. *See* EX1001, 1:26-28

(“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).

71. 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.

72. 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

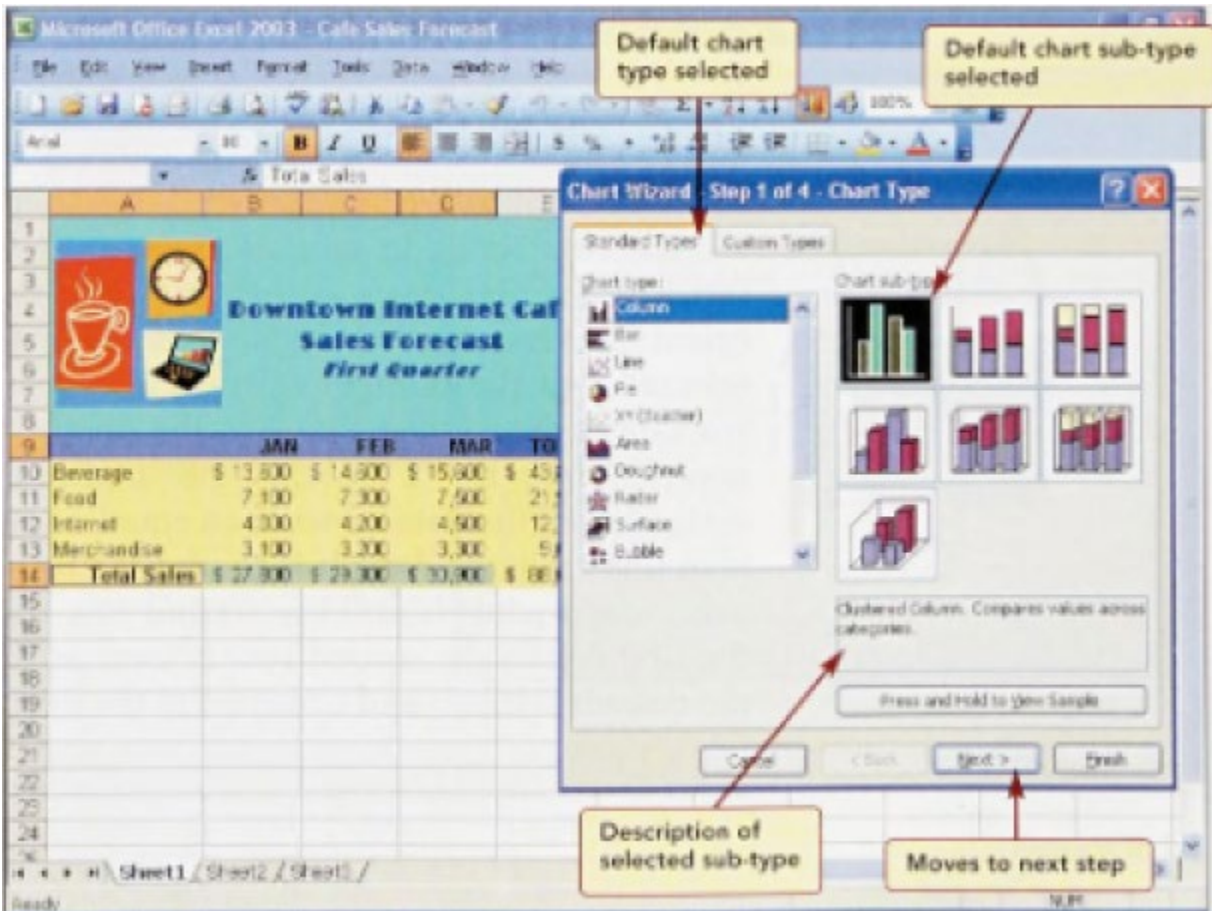
3 A chart is a visual representation of data that is used to convey information in an easy-to-understand and attractive manner. Different types of charts are used to represent data in different ways. The type of chart you create depends on the type of data you are charting and the emphasis you want the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

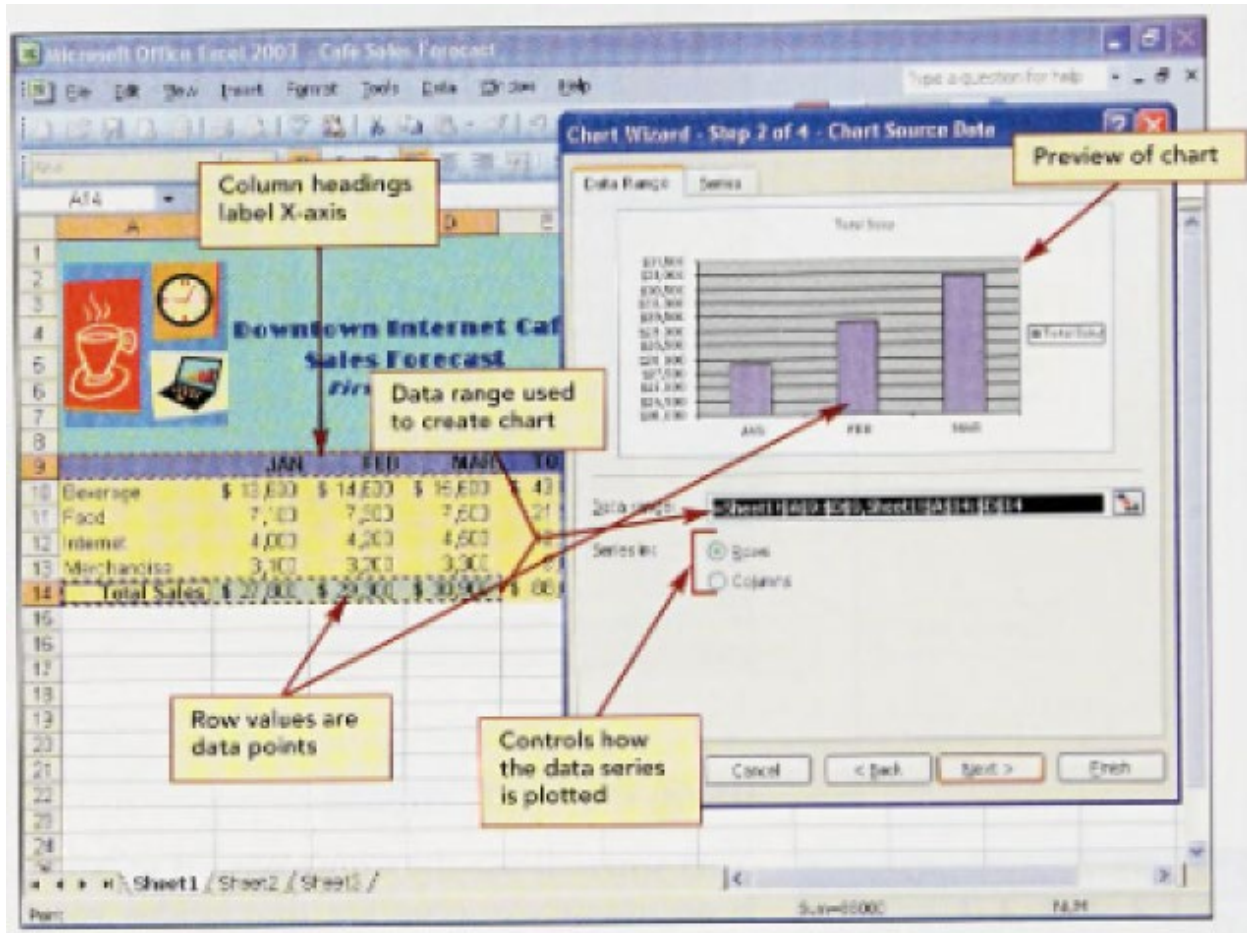
EX1011, 0079.

73. 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.



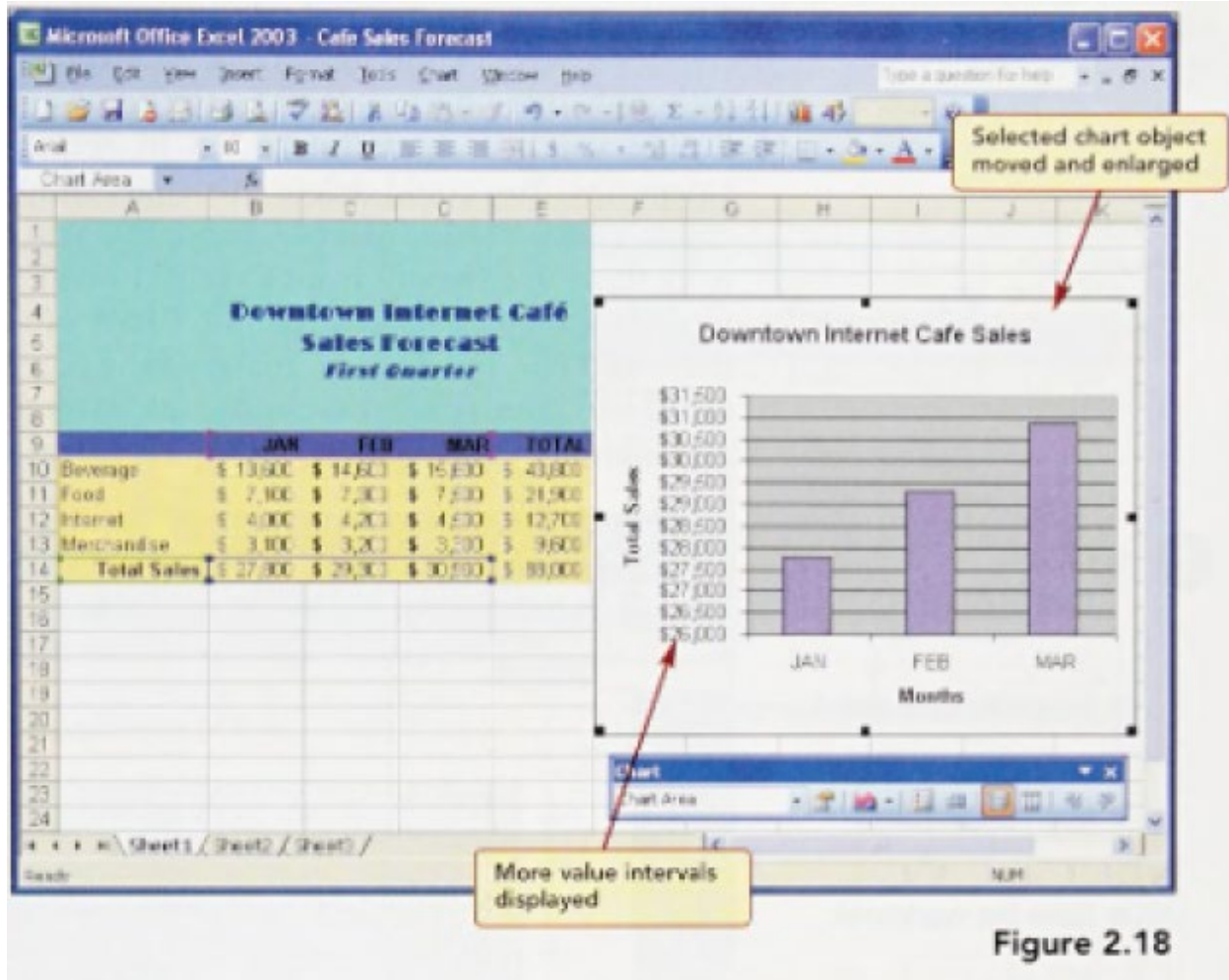
EX1011, 82 (Figure 2.11).

74. 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.



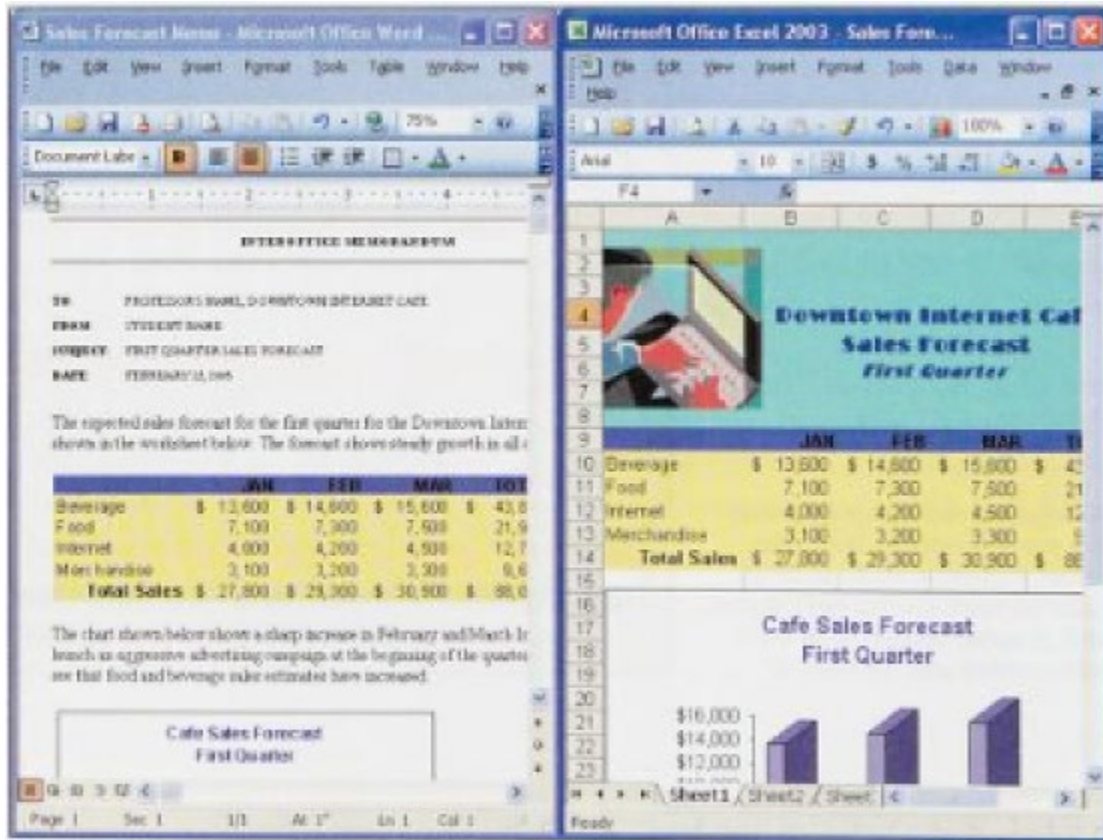
EX1011, 0083 (Figure 2.12).

75. 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).

76. 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.

77. 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.

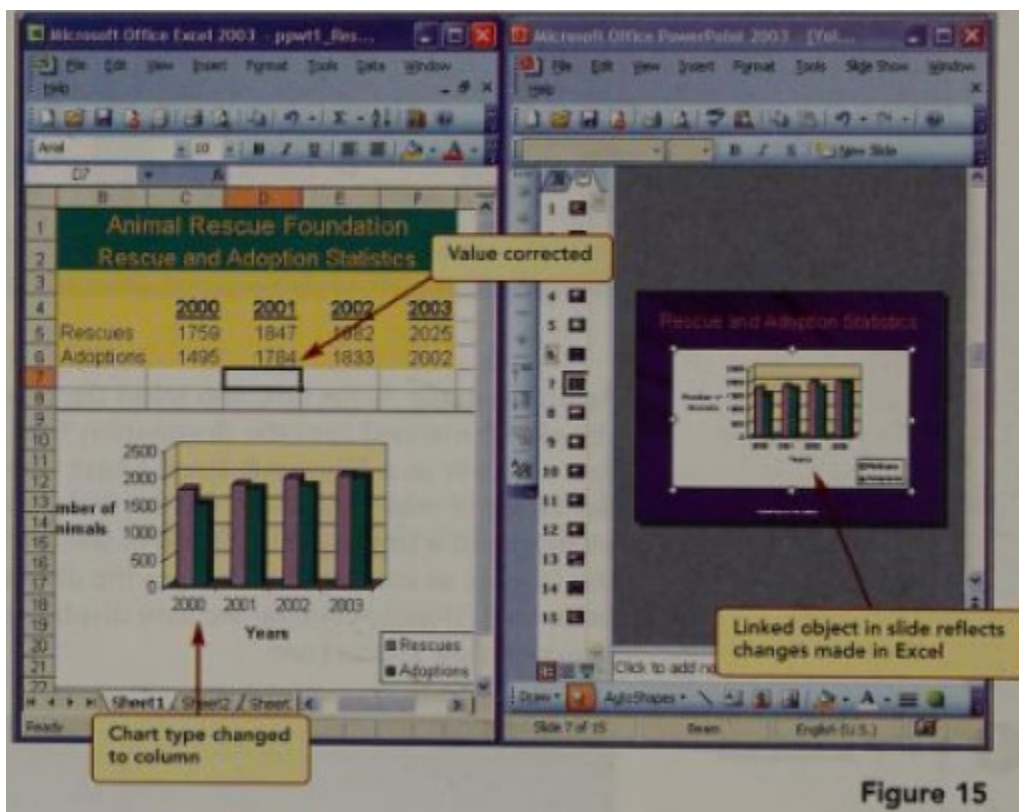


Figure 15

EX1012, 0025 (Figure 15).

78. 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.

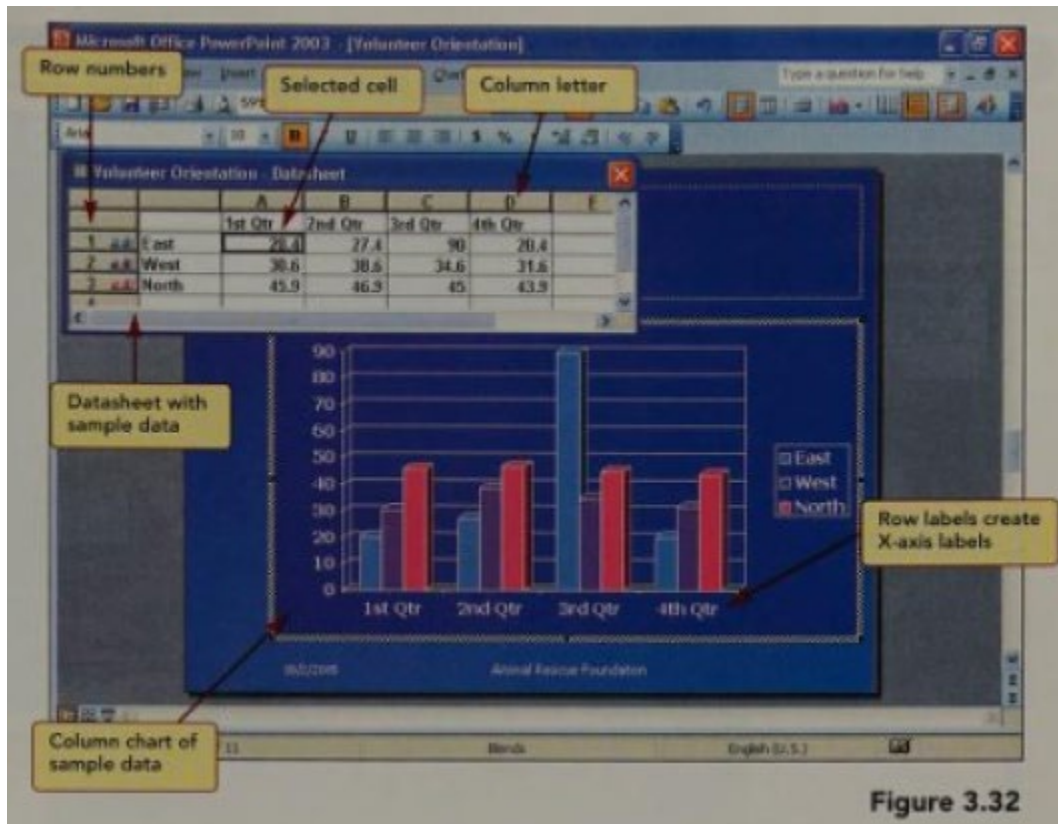
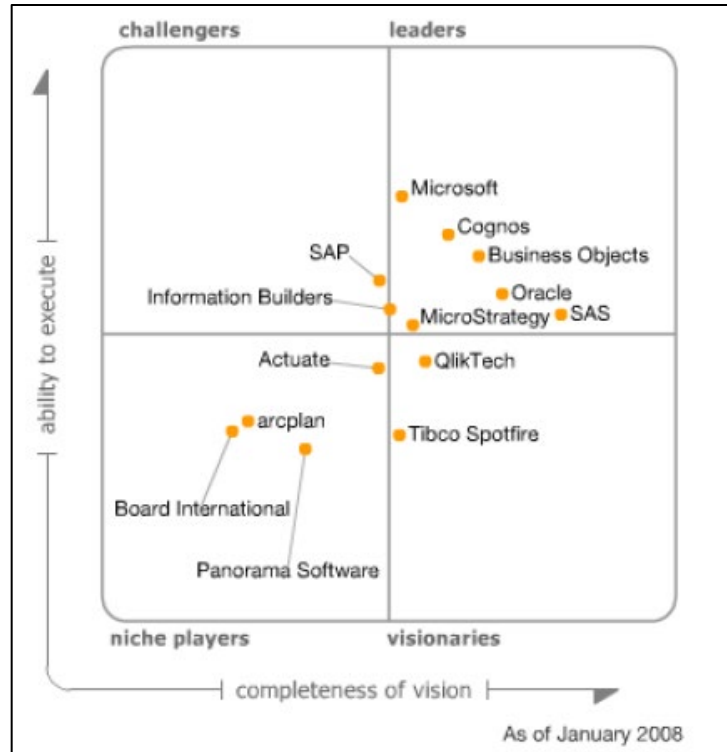


Figure 3.32

EX1012, 68 (Figure 3.32).

79. 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.

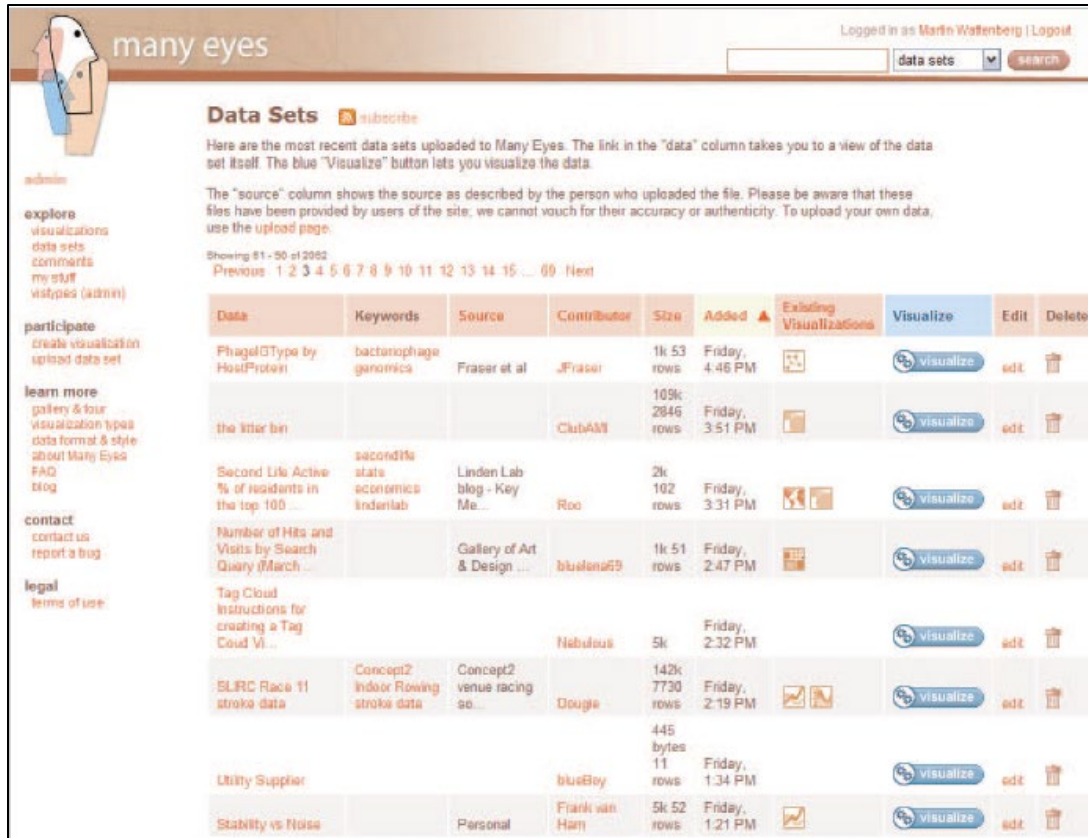


EX1028, 0002 (Figure 1).

80. 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.

81. 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.

82. 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:



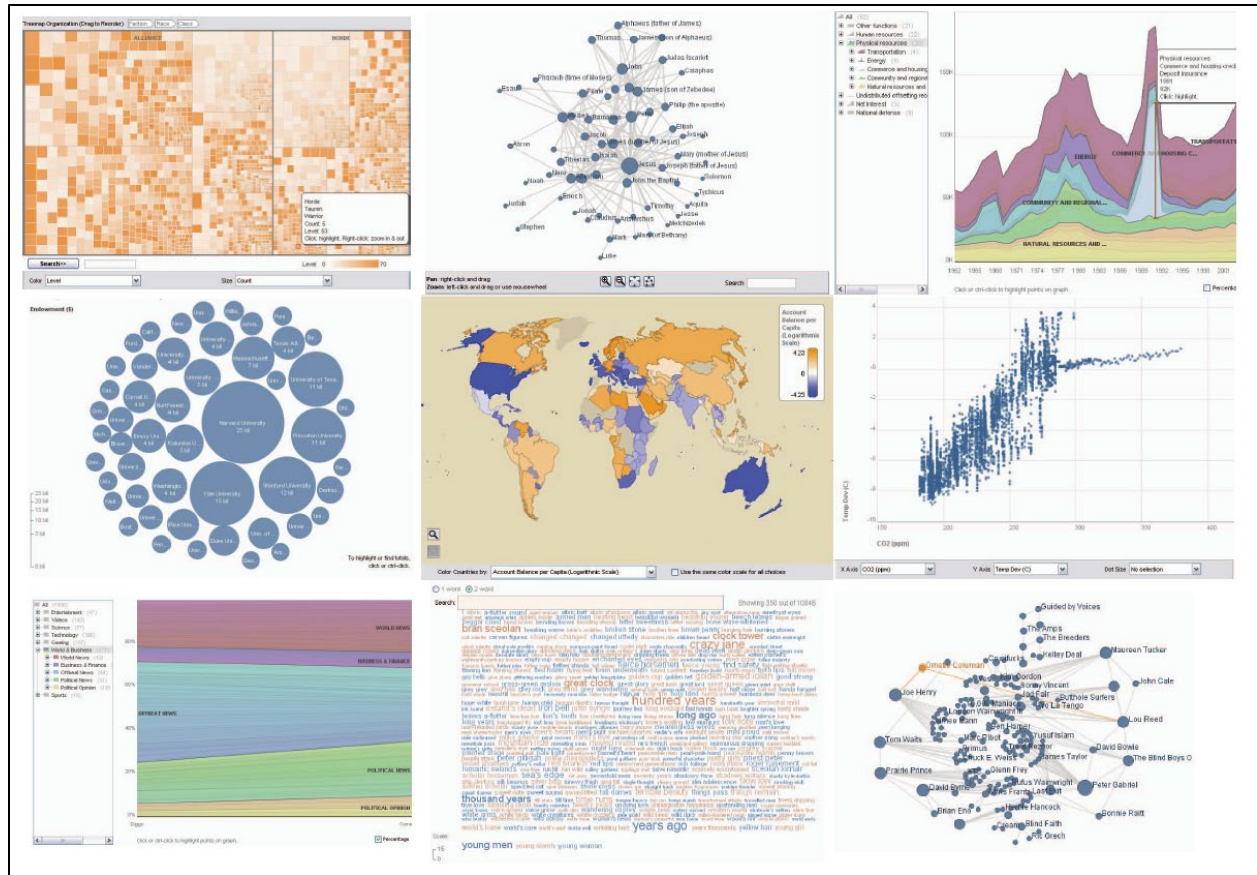
The screenshot shows the ManyEyes website interface. At the top, there is a header with the ManyEyes logo, a search bar, and a login status for 'Martin Wallenberg'. Below the header, there is a sidebar with navigation links: 'admin', 'explore' (visualizations, data sets, comments, my stuff, visit page (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	Excluding 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
SLIRC 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).

83. 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.



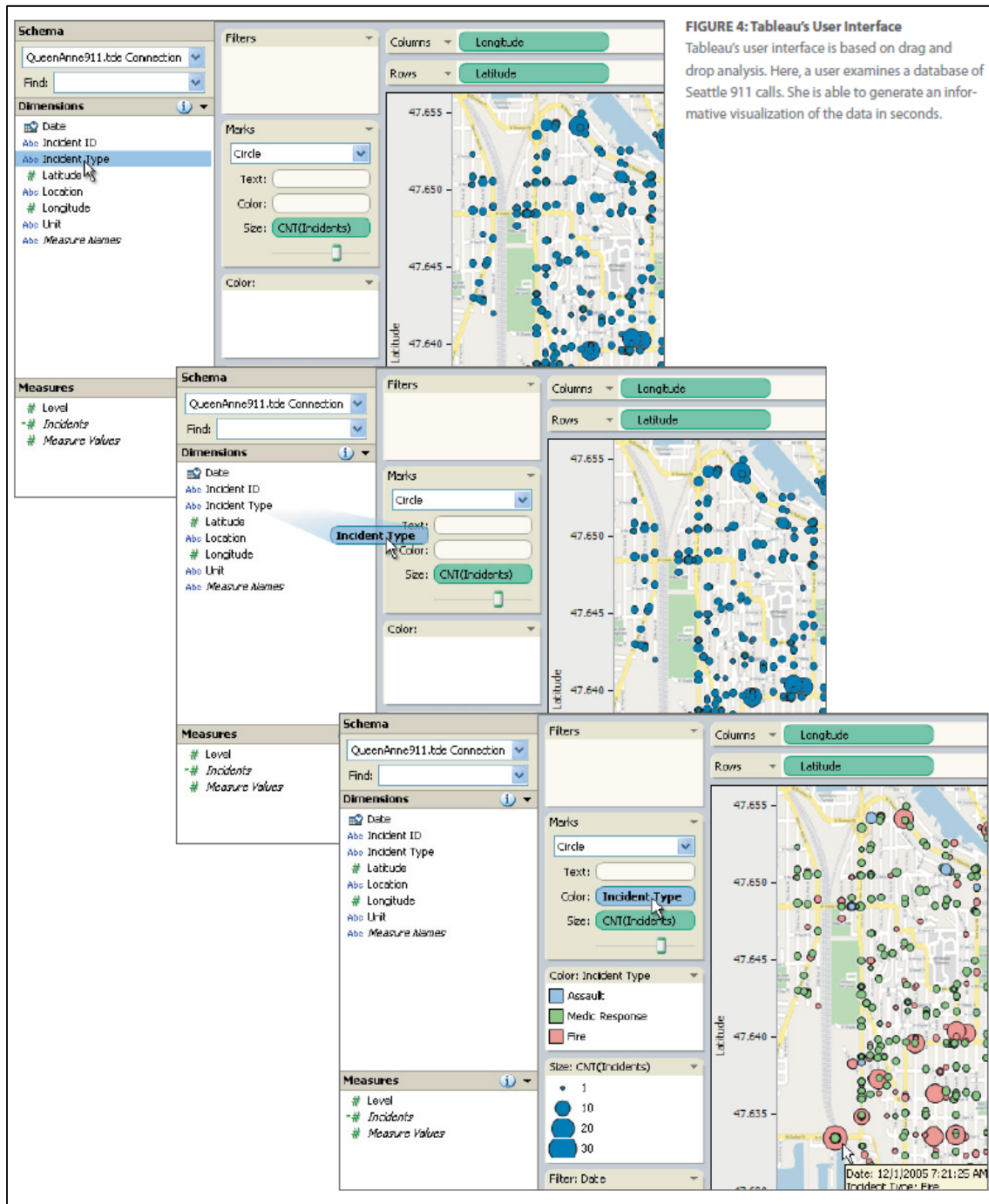
EX1039, 0004 (Figure 2).

84. 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.

85. 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

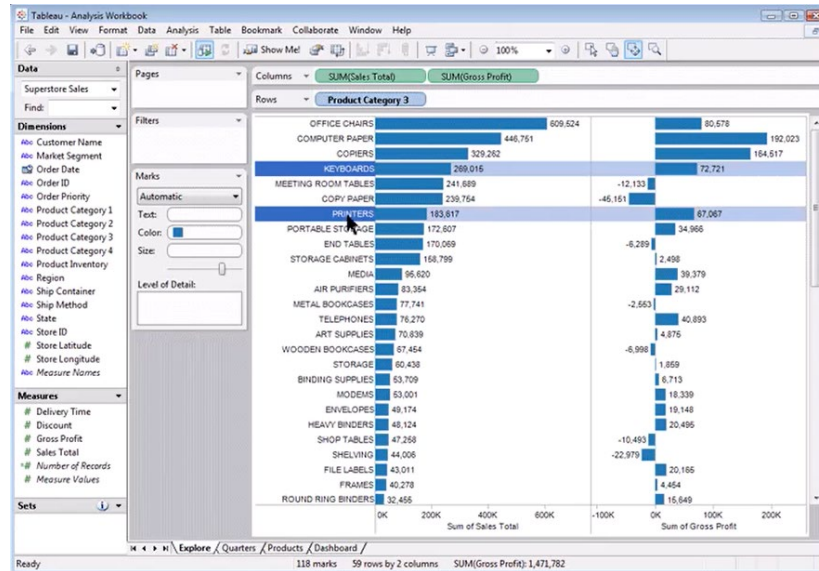
interface). Mouse-over pop-up windows were known. *See* EX1035, 0010 (discloses showing labels when the mouse is moved over corresponding data); EX1053 (incorporating mouse over pop-ups in an interactive graphic). Drill down features were known. *See* EX1039, 0007 (discloses interactive features allowing 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).

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



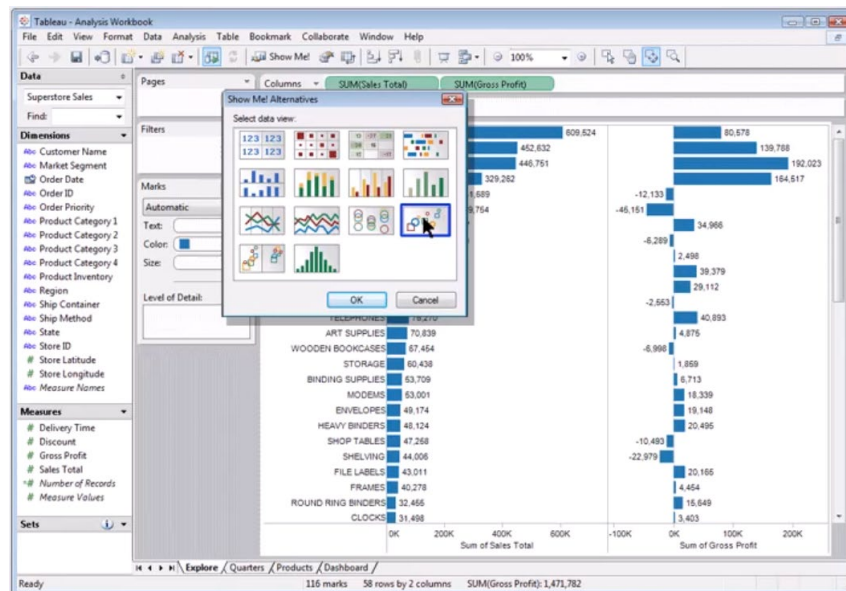
EX1041, 0012; see also EX1049.

87. 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.

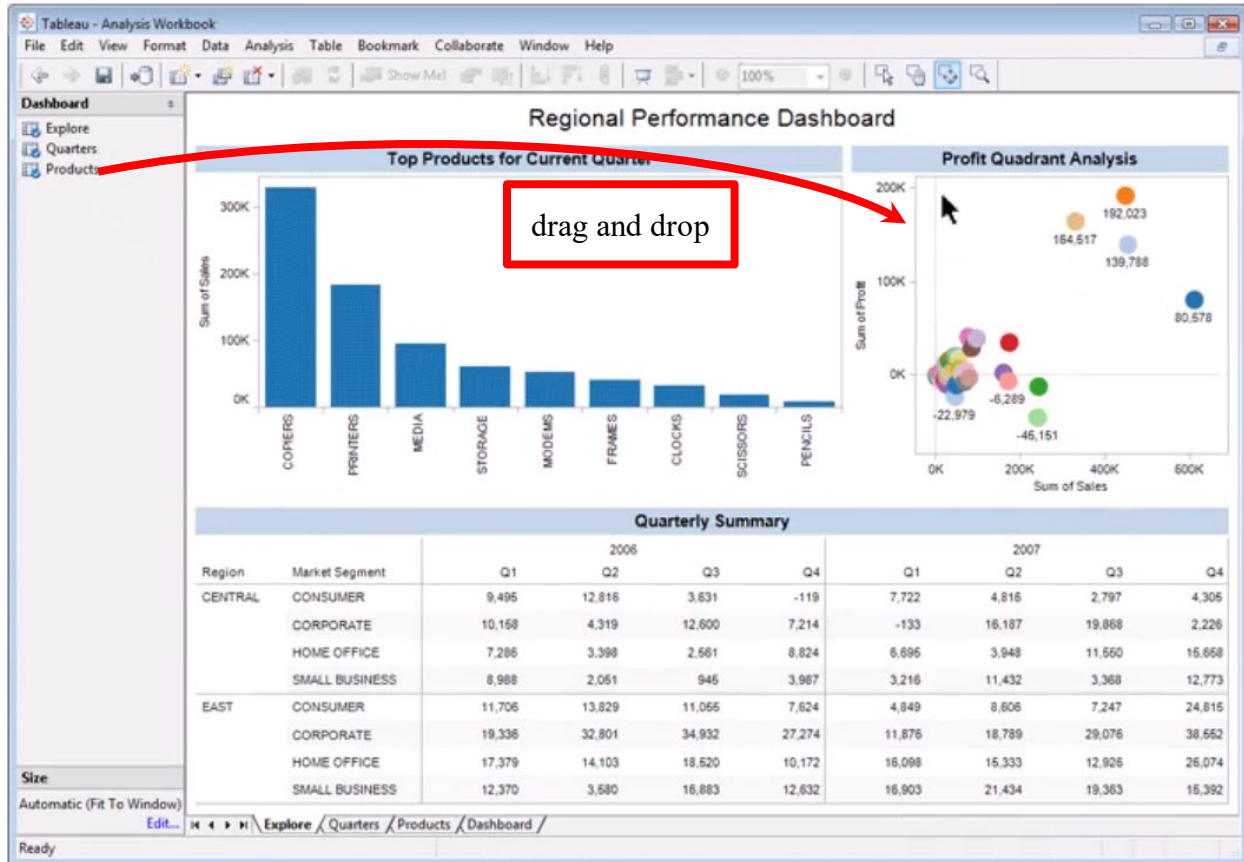
88. Shown below, after selecting which data the user wants to see, the user can select different chart types to visualize the data differently.



EX1049, 00:50.

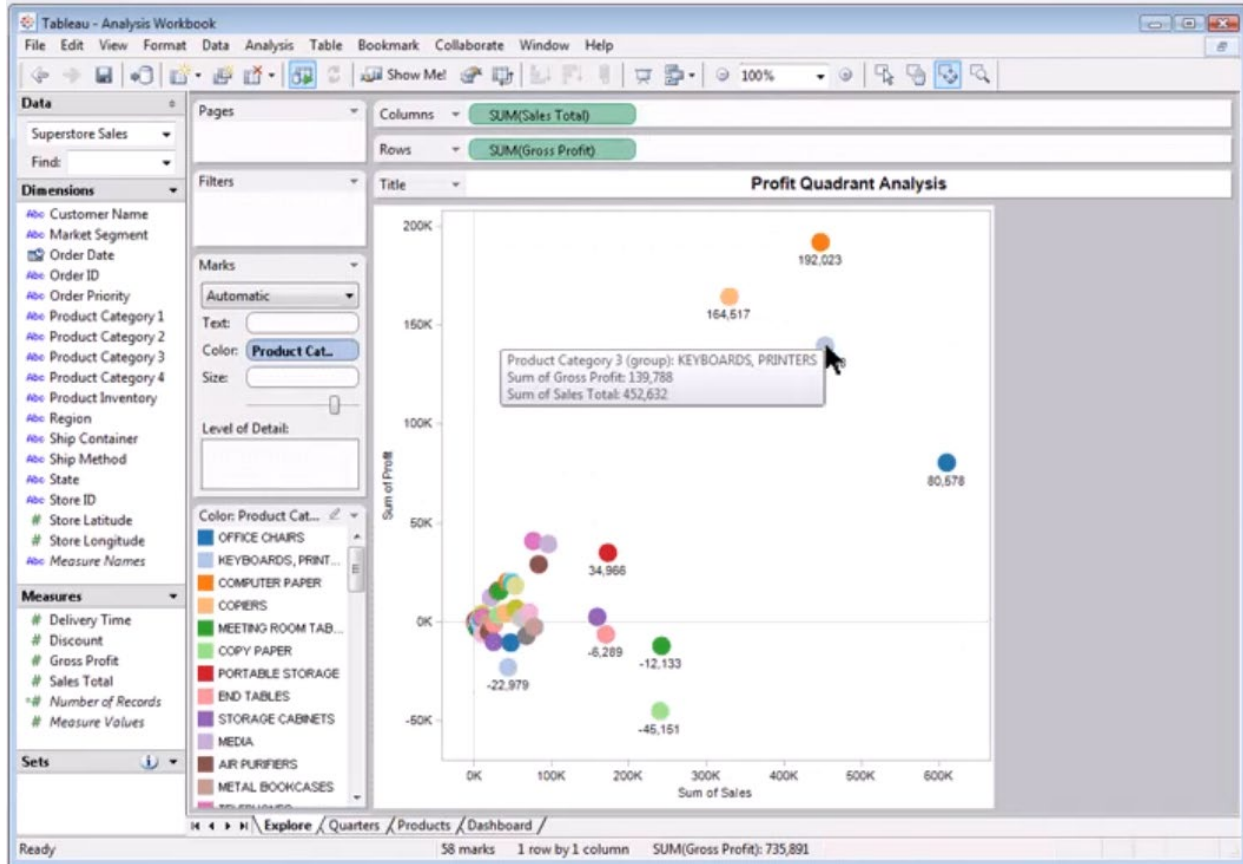
89. Tableau users could also add charts to dashboards for ongoing use.

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).

90. 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.

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

B. Flash

92. 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.

93. 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.

94. 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.

95. 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).

96. 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

97. 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.

98. 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.

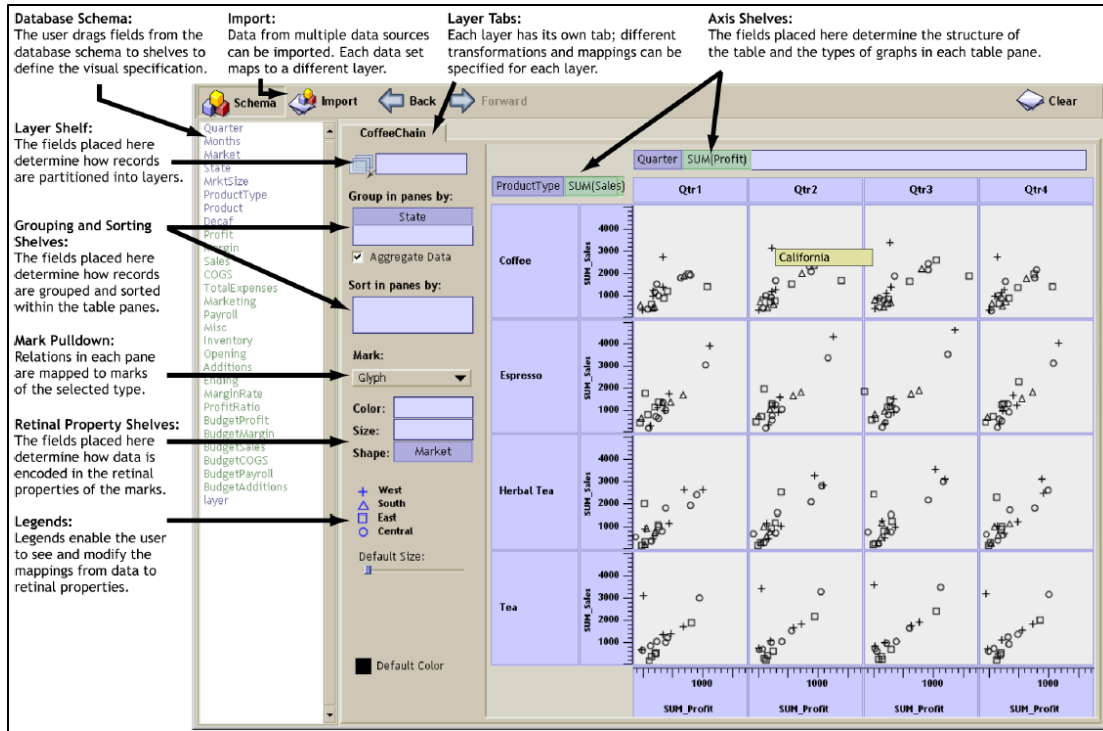
99. JavaScript can also be used within a framework known as asynchronous JavaScript and XML (AJAX). EX1057, 0025. In AJAX, the browser submits a request for content from the server using an XML HTTP request. EX1057, 0040, 0043. The developer then defines a function to call when the response is received. 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

100. 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.

101. 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).

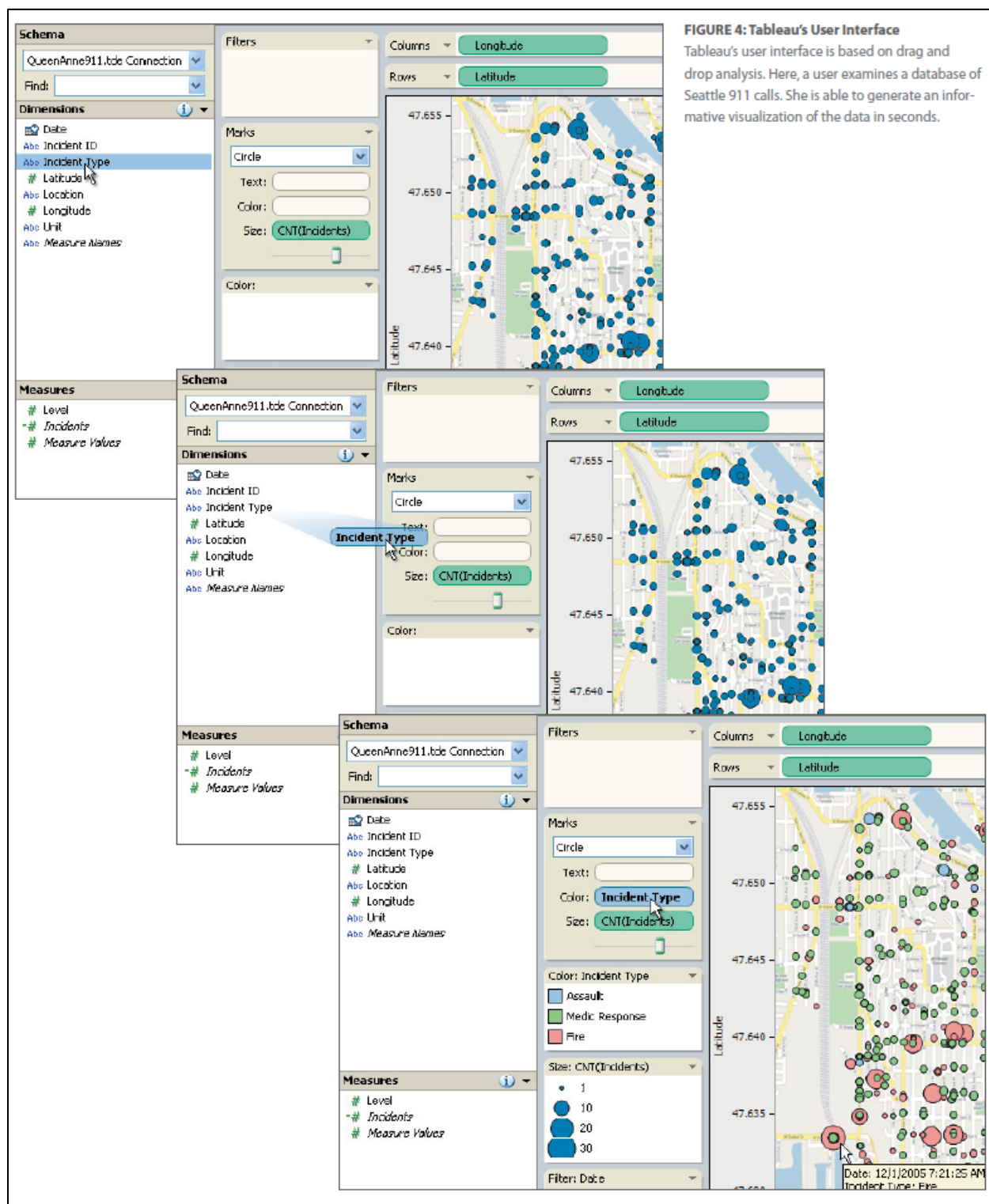
102. Another system called Polaris used drag and drop to “construct table-based displays of relational data.” EX1040, 0003. Shown below, “[t]he user drags fields from the database schema to shelves to define the visual specification.” EX1040, 0003 (Fig. 1).



EX1040, 0003 (Fig. 1).

103. 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.

104. Tableau also used drag-and-drop:



EX1041, 0012; *see also* EX1049.

105. 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.

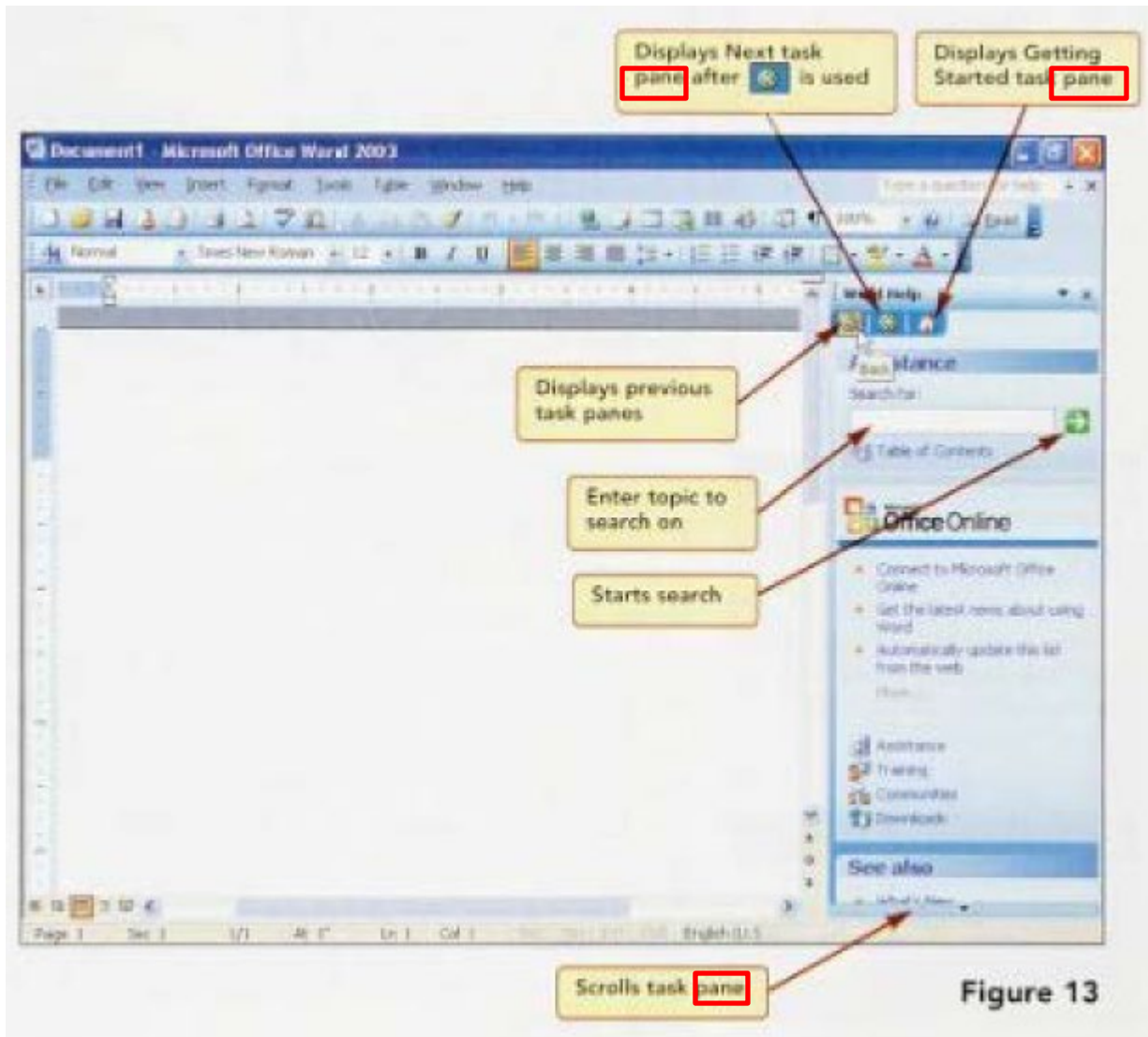
106. By 2008, drag-and-drop had become a basic, well-known design choice in graphical user interfaces.

E. Panes

107. 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.

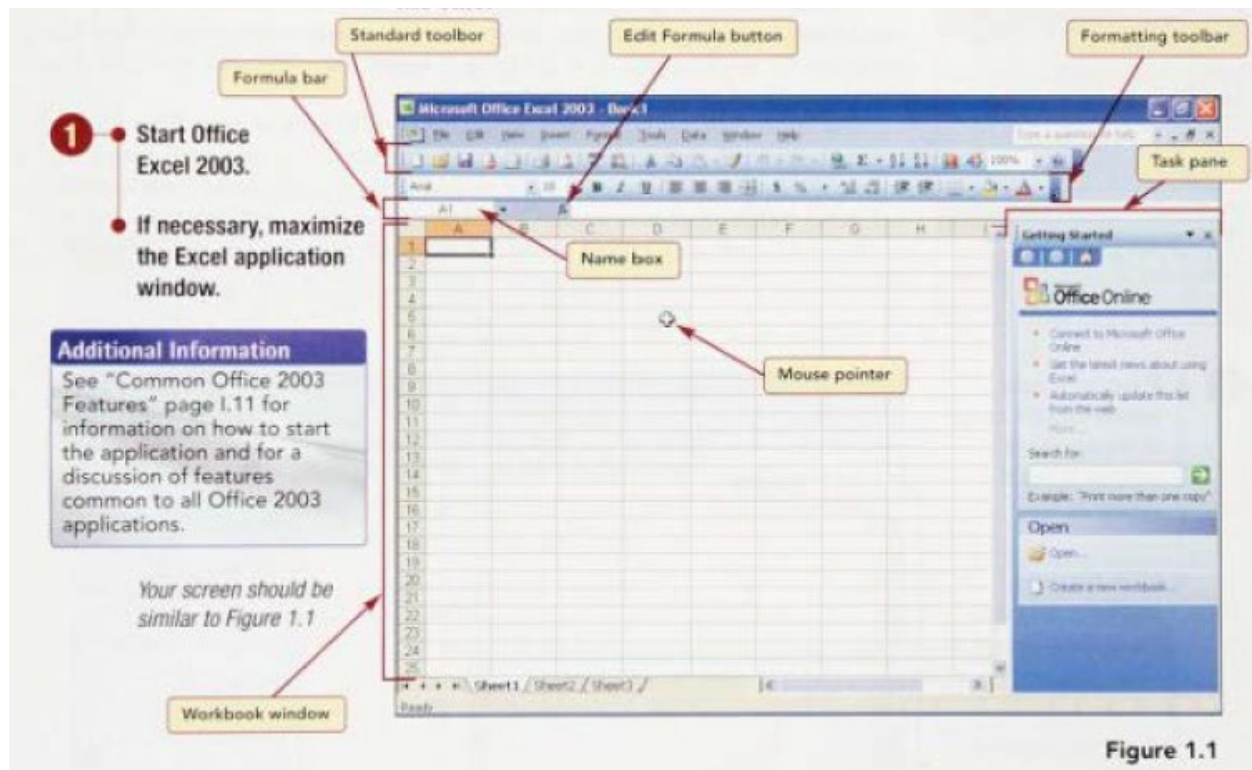
108. Panes are common interface components used to organize a layout. For example, an email client such as Microsoft Outlook has multiple panes. Here, panes may be used to display available folders, items in a selected folder, a selected item, and upcoming calendar events.

109. 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:



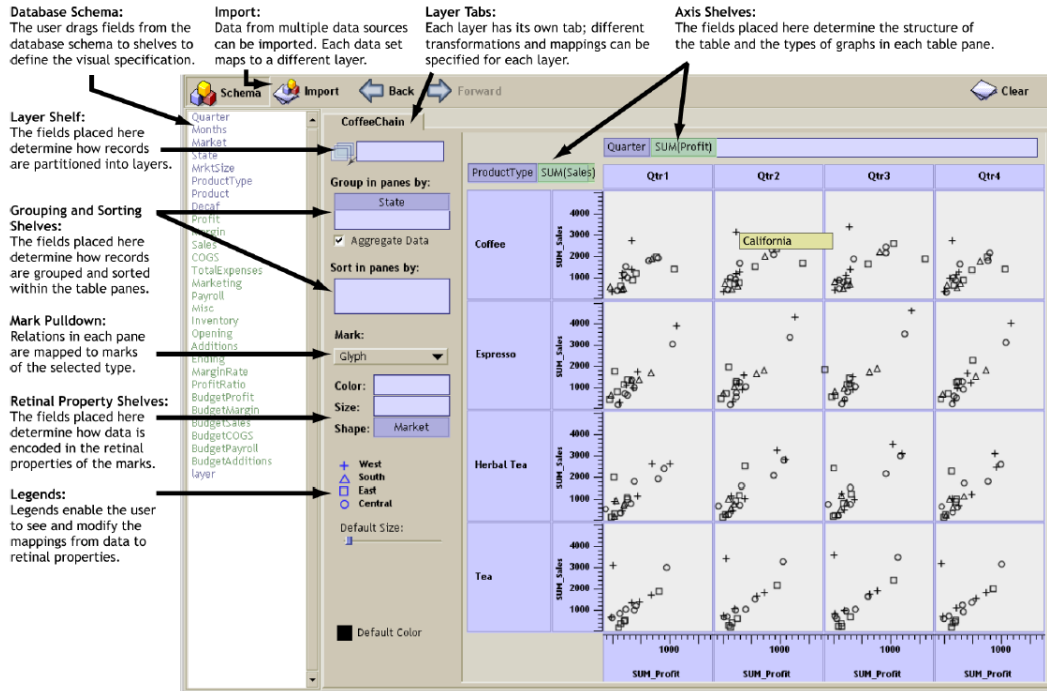
EX1011, 0034 (Figure 13) (annotated).

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



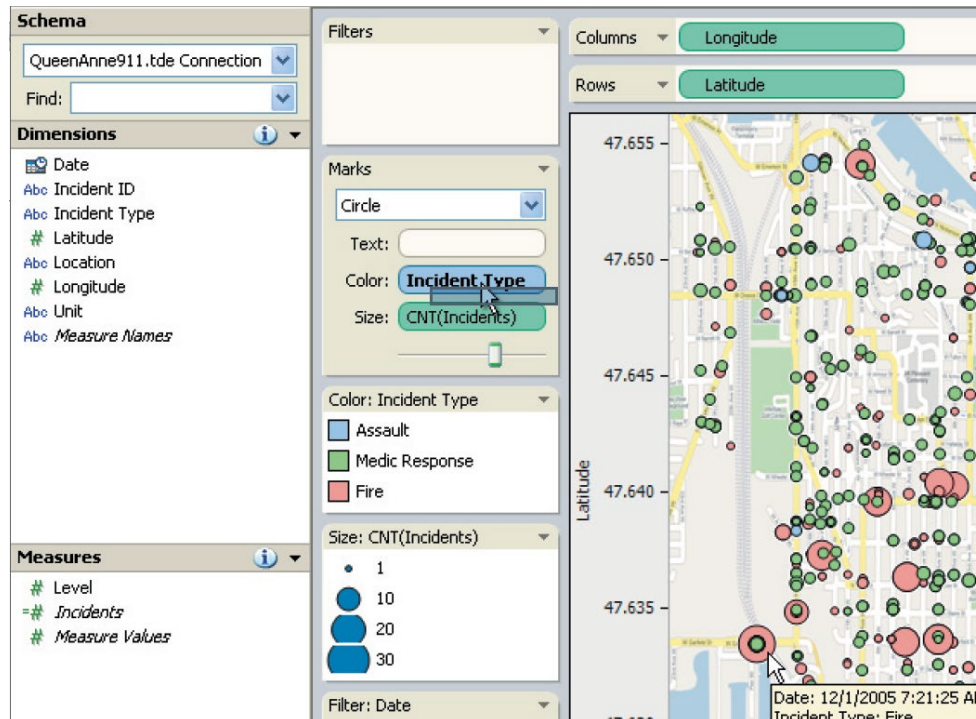
EX1011, 0044 (Figure 1.1).

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



EX1040, 0003.

112. Tableau also used panes:



EX1041, 0012; see also EX1049.

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

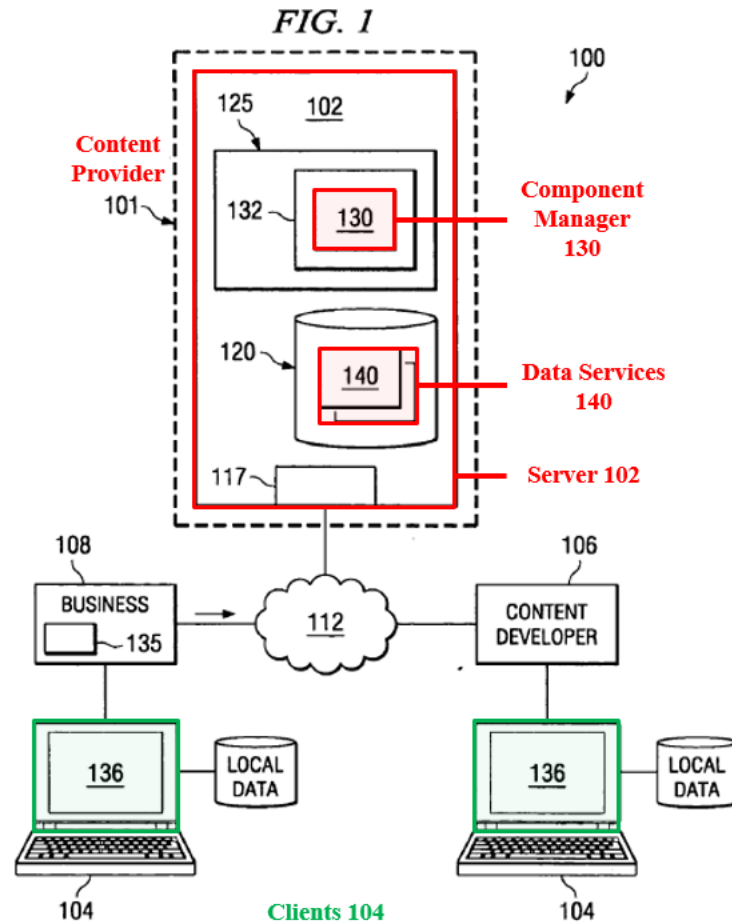
VII. Overview of the Prior Art

114. This section provides an overview of the prior art and demonstrates that displaying an interactive chart on a second website was well known in the industry well before the priority date of the '595 patent.

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

115. 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.

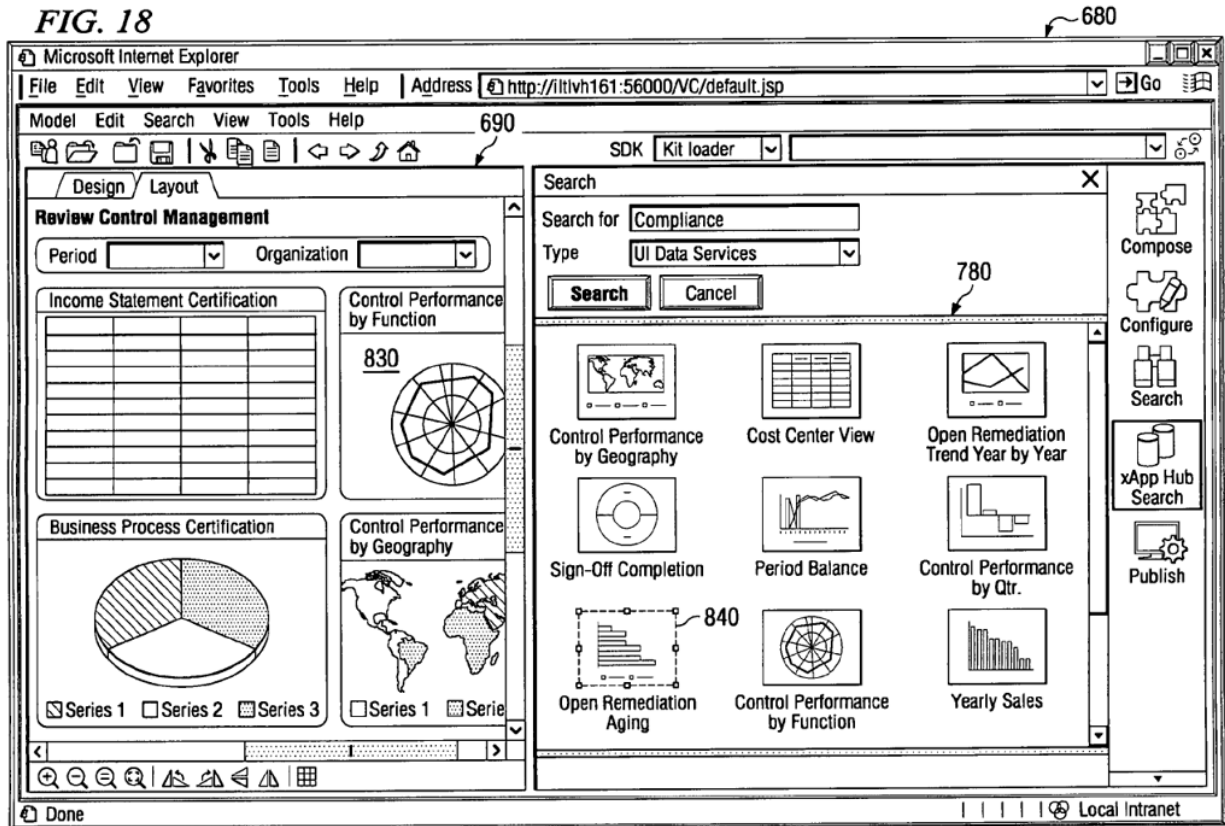
116. Figure 1 of Rostoker “illustrates an example system 100 for presenting, providing, or otherwise managing enterprise application content and other data services 140 via a content provider 101.” EX1005, ¶18. As illustrated below, “system 100” includes “server 102, one or more clients 104, one or more 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).

117. 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

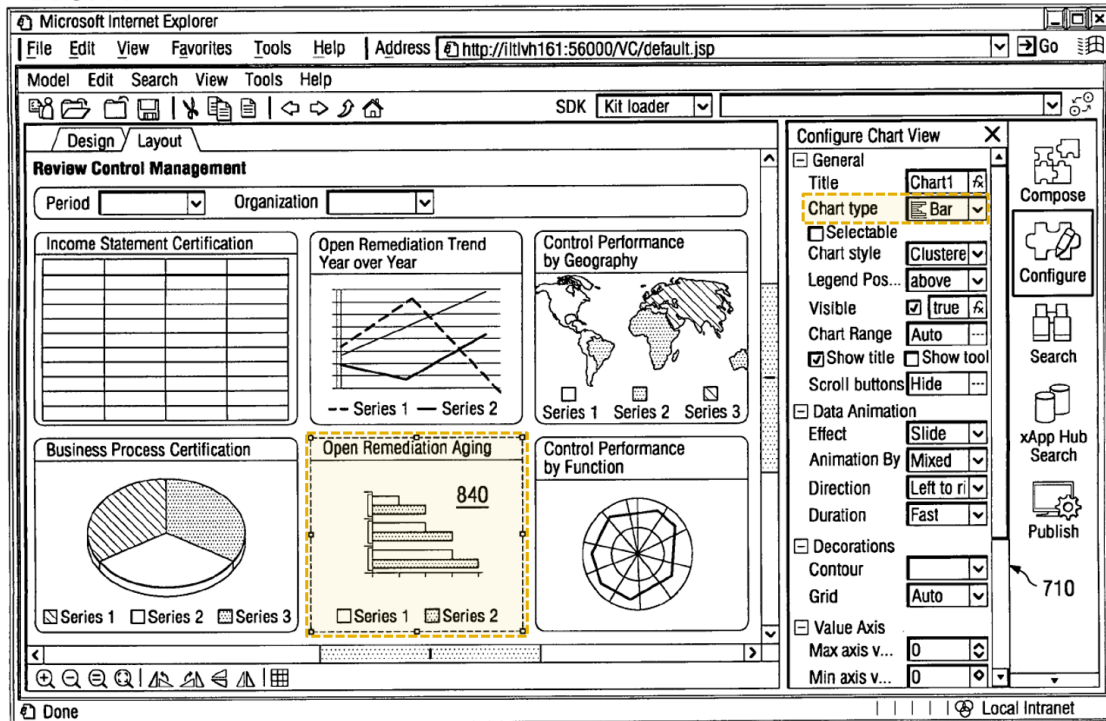
130) for creating a custom application, such as a business application, to fulfill a need of the user,” as illustrated, for example, in Figure 18. EX1005, ¶40.



EX1005, FIG. 18.

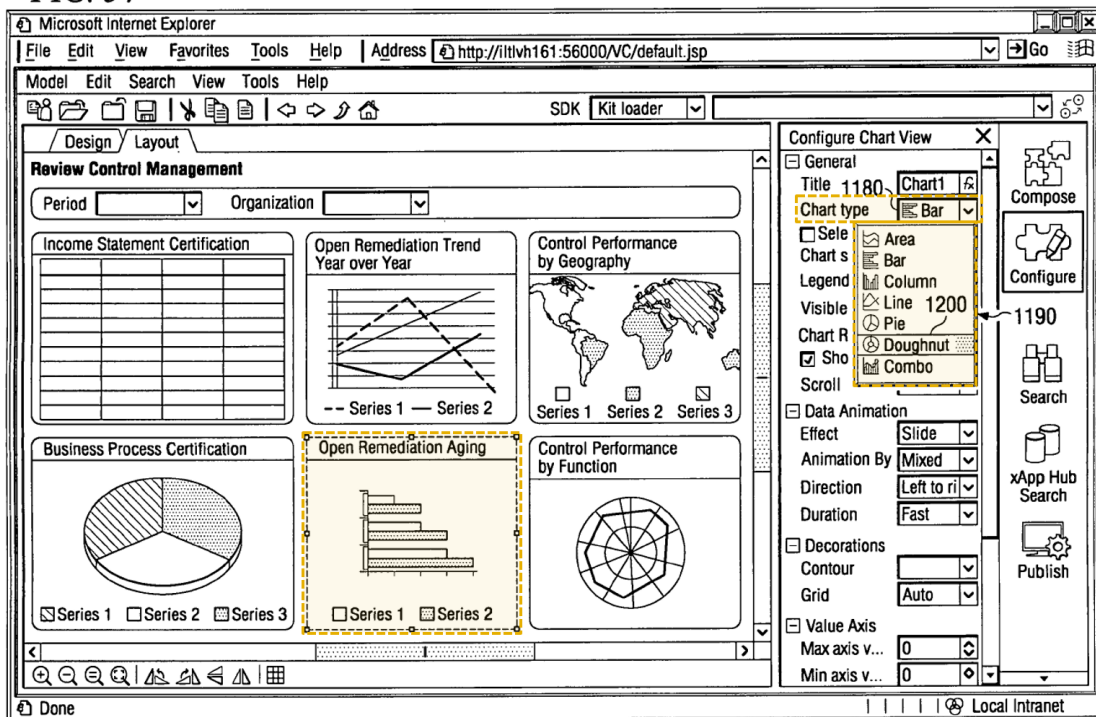
118. Figures 53-55 below show an example of configuring an xApp 840. Figure 53 shows the configure function for xApp 840 is selected, Figure 54 shows the user changes the chart type of the xApp 840 by selecting the desired type 1190 (e.g., “Doughnut” chart type 1200) from the list 1180, and Figure 55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

FIG. 53



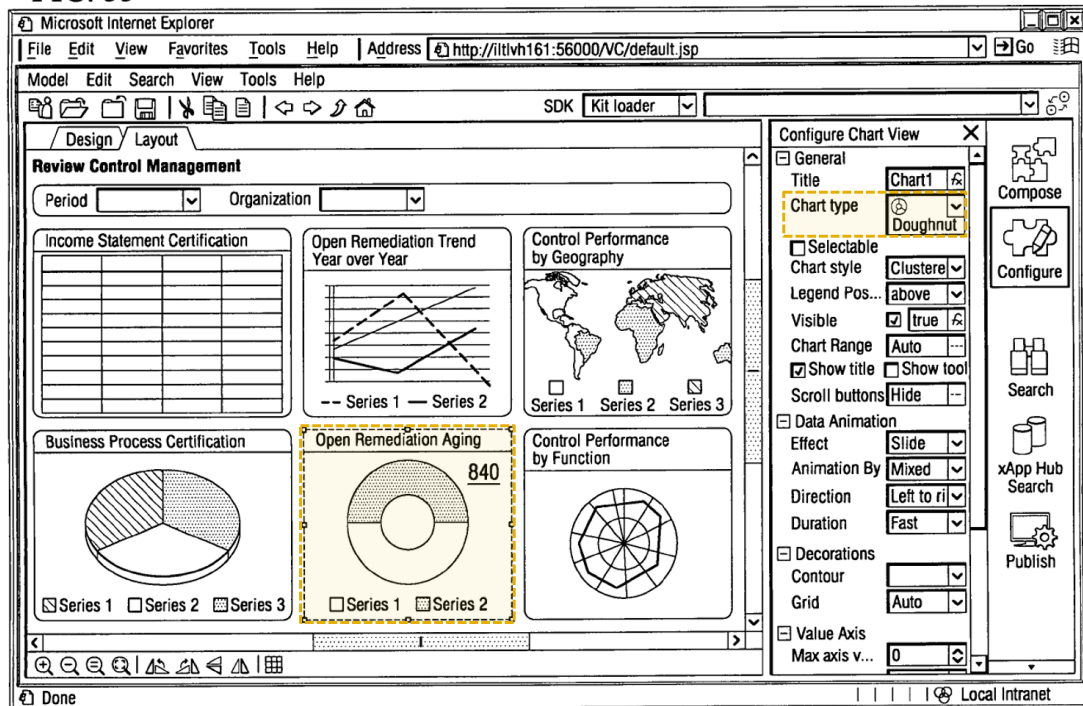
EX1005, FIG. 53 (annotated).

FIG. 54



EX1005, FIG. 54 (annotated).

FIG. 55

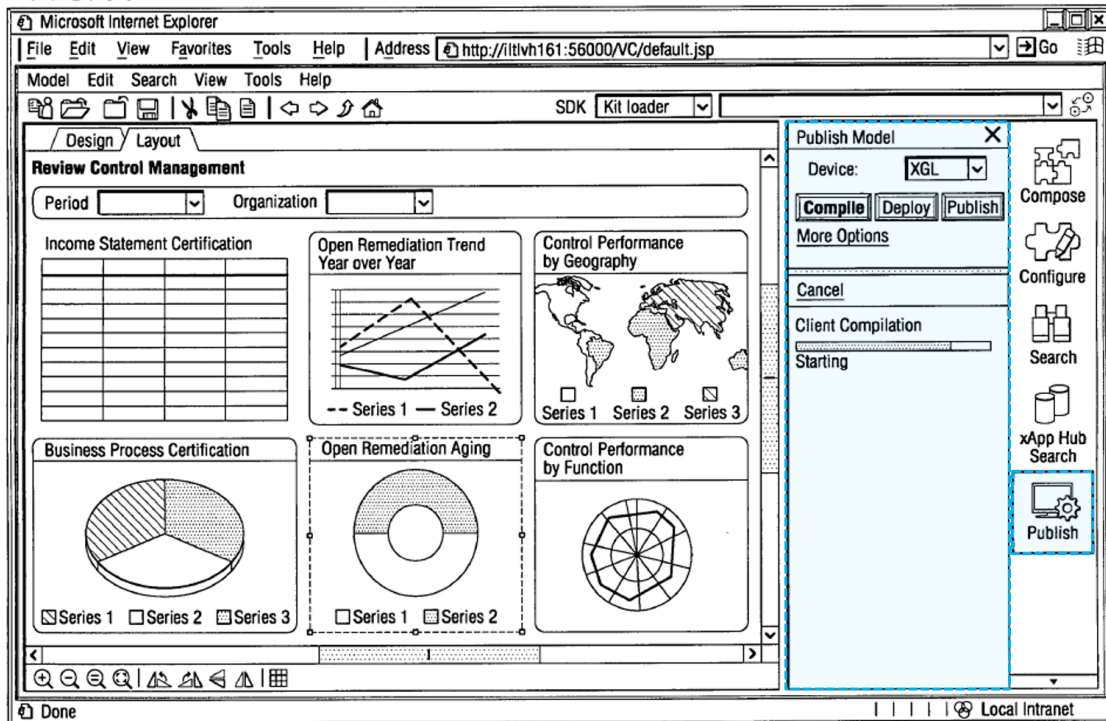


EX1005, FIG. 55 (annotated).

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

120. 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.

121. 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”). 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.

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

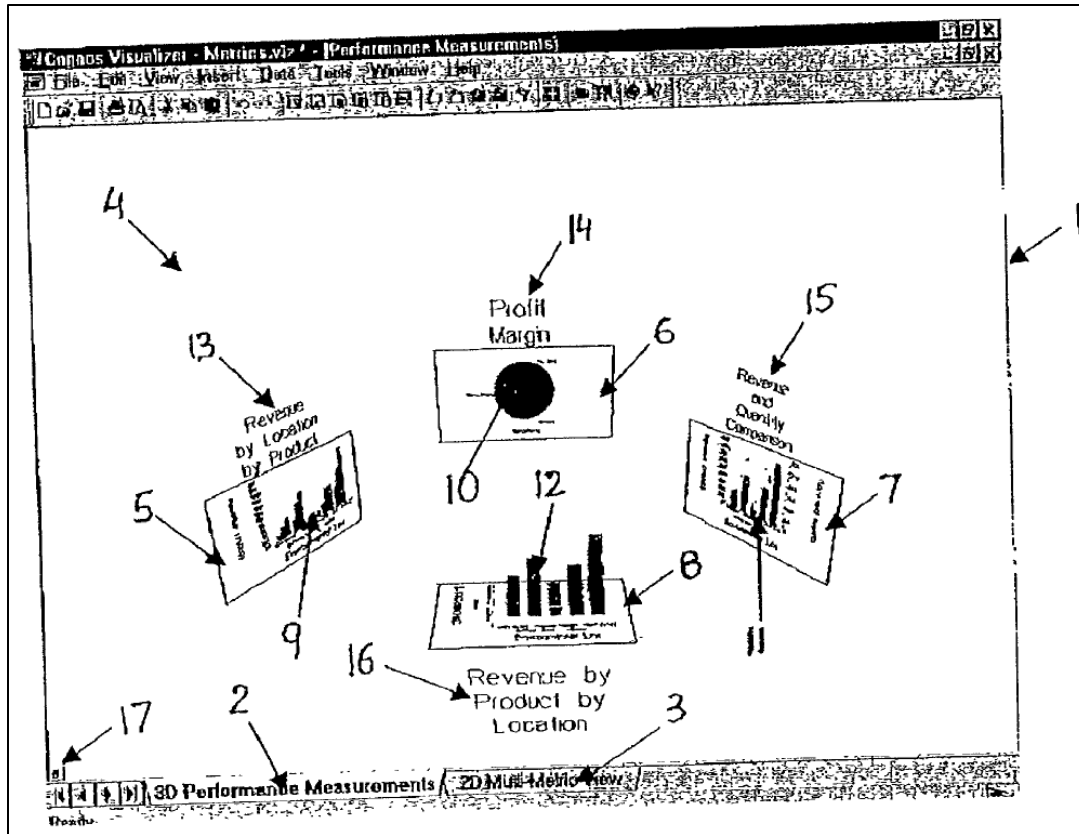
122. Jou “provides an interactive system for visualizing business data.”

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.

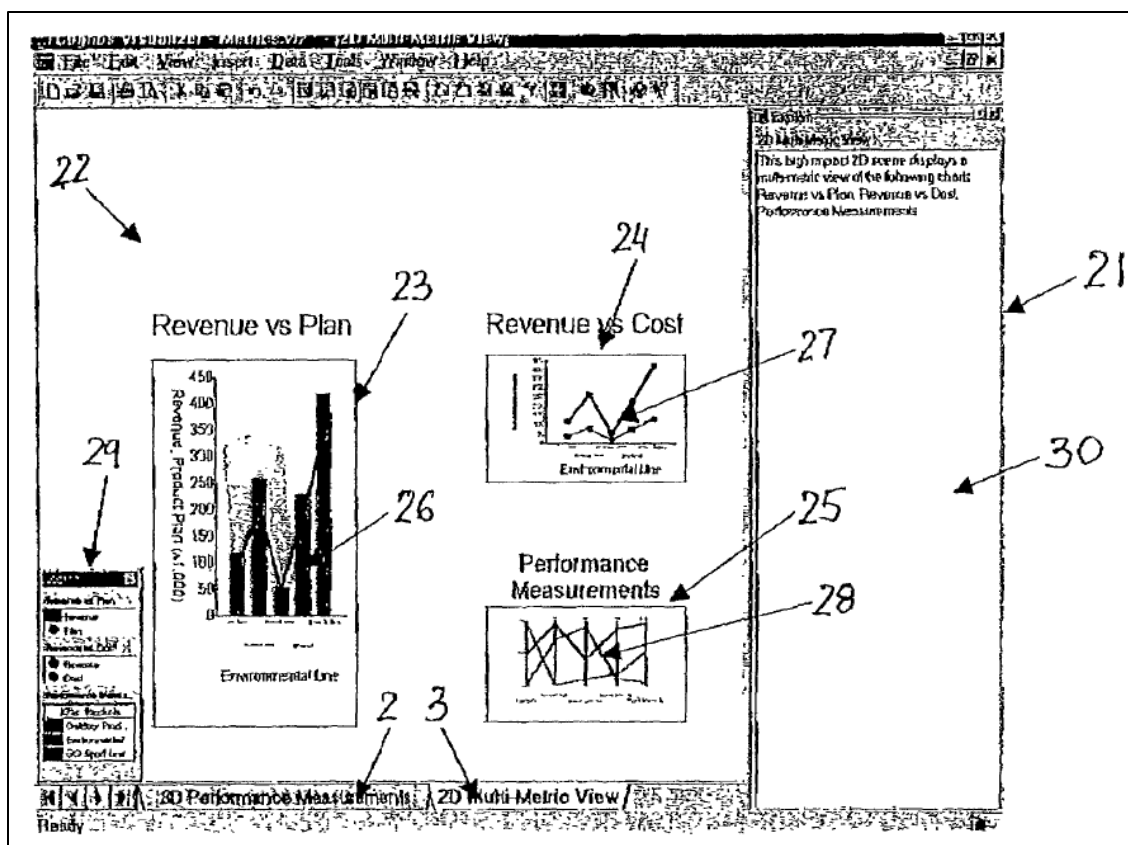
123. “A visualization . . . can be built . . . from a blank visualization and adding manually various visualization elements” EX1007, ¶13. Jou provides “a large variety of predefined types of two- and three-dimensional charts, such as line, bar, surface, pie, swatch, and map charts.” EX1007, ¶¶14, 36.

124. 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.

125. 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.



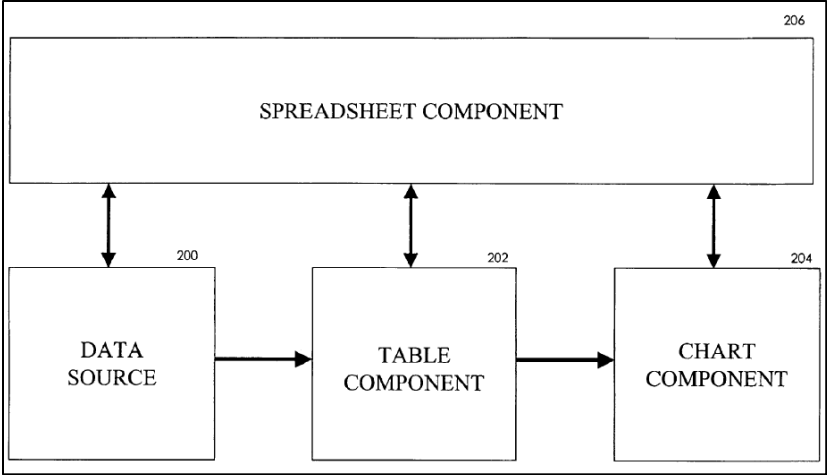
EX1007, FIG. 2B.

126. 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.

127. 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. Couckuyt discloses a tool with a chart pane and data pane used to design interactive charts with drag-and-drop.

128. Couckuyt discloses a “drag-and-drop user interface (DDUI) for generating a chart report having a single chart.” EX1006, 7:10-11. “Dropping a field icon [from a field list] into a drop zone causes the charting DDUI to populate a corresponding portion of the chart with the data from the field.” EX1006, 1:32-34. As depicted below, Couckuyt discloses a “spreadsheet component . . . for creating or modifying an electronic spreadsheet document” EX1006, 4:26-28. Chart component 204 may be used to render a chart or graph from data at data source 200 and/or spreadsheet component 206. EX1006, 4:27-31. Table component 202 is a module to create interactive tables summarizing data from data source 200. EX1006, 4:48-50. An example table component may be a pivot table within Microsoft Excel. EX1006, 4:46-49.



EX1006, FIG. 2.

129. Figure 3A depicts data source 300 maintained in a spreadsheet document. EX1006, 6:1-3.

A spreadsheet table with 13 rows and 6 columns. The columns are labeled A through E. The first row contains headers: Month, Product, Salesperson, Sales, and Region. The subsequent rows contain data for months May and June, listing products (Dairy, Meat), salespersons (Davolio, Buchanan), sales amounts, and regions (West, South, East, North). An arrow points from the label '300' to the table.

	A	B	C	D	E
1	Month	Product	Salesperson	Sales	Region
2	May	Dairy	Davolio	\$5,477	West
3	May	Meat	Buchanan	\$8,012	East
4	May	Dairy	Davolio	\$9,411	West
5	May	Dairy	Davolio	\$6,089	South
6	May	Meat	Buchanan	\$9,566	South
7	Jun	Dairy	Davolio	\$6,805	East
8	Jun	Meat	Buchanan	\$6,085	South
9	Jun	Meat	Buchanan	\$265	East
10	Jun	Meat	Davolio	\$5,575	South
11	Jun	Dairy	Buchanan	\$5,971	North
12	Jun	Dairy	Buchanan	\$4,046	West
13	Jun	Meat	Buchanan	\$1,361	North

EX1006, FIG. 3A.

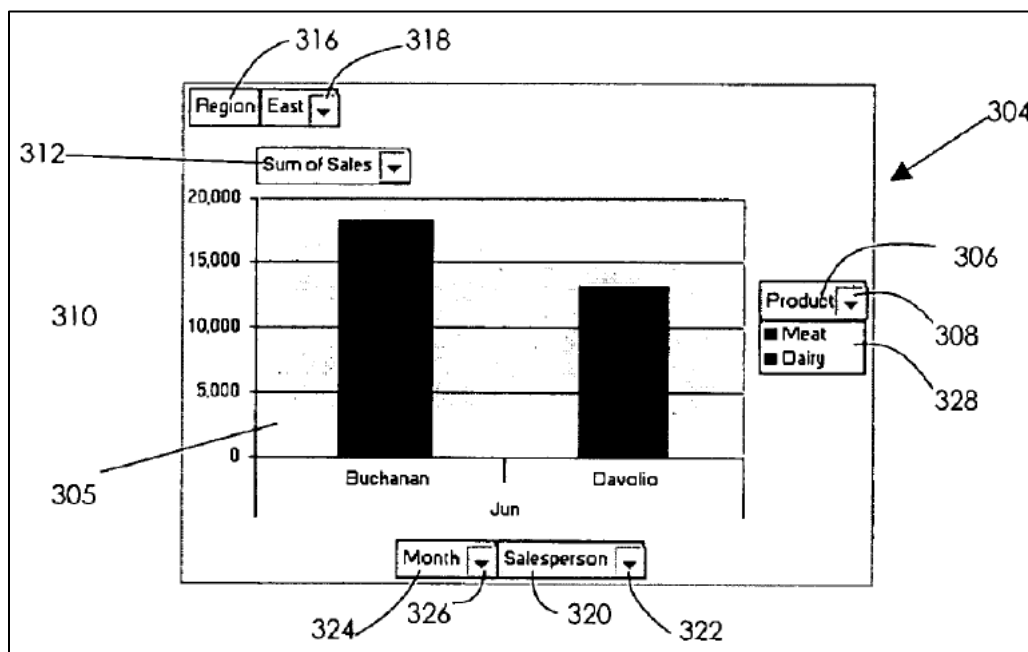
130. Figure 3B depicts a data source including “table report 302 in connection with the associated source data 303.” EX1006, 6:8-10.

Source data				PivotTable report				
	A	B	C	D	E	F	G	H
1	Sport	Quarter	Sales					
2	Golf	Qtr3	\$1,500					
3	Golf	Qtr4	\$2,000		Sum of Sales	Quarter ▼		
4	Tennis	Qtr3	\$600		Sport ▼	Qtr3	Qtr4	Grand Total
5	Tennis	Qtr4	\$1,500		Golf	\$7,930	\$2,000	\$9,930
6	Tennis	Qtr3	\$4,070		Tennis	\$4,670	\$6,500	\$11,170
7	Tennis	Qtr4	\$5,000		Grand Total	\$12,600	\$8,500	\$21,100
8	Golf	Qtr3	\$6,430					

Source values for cell F5

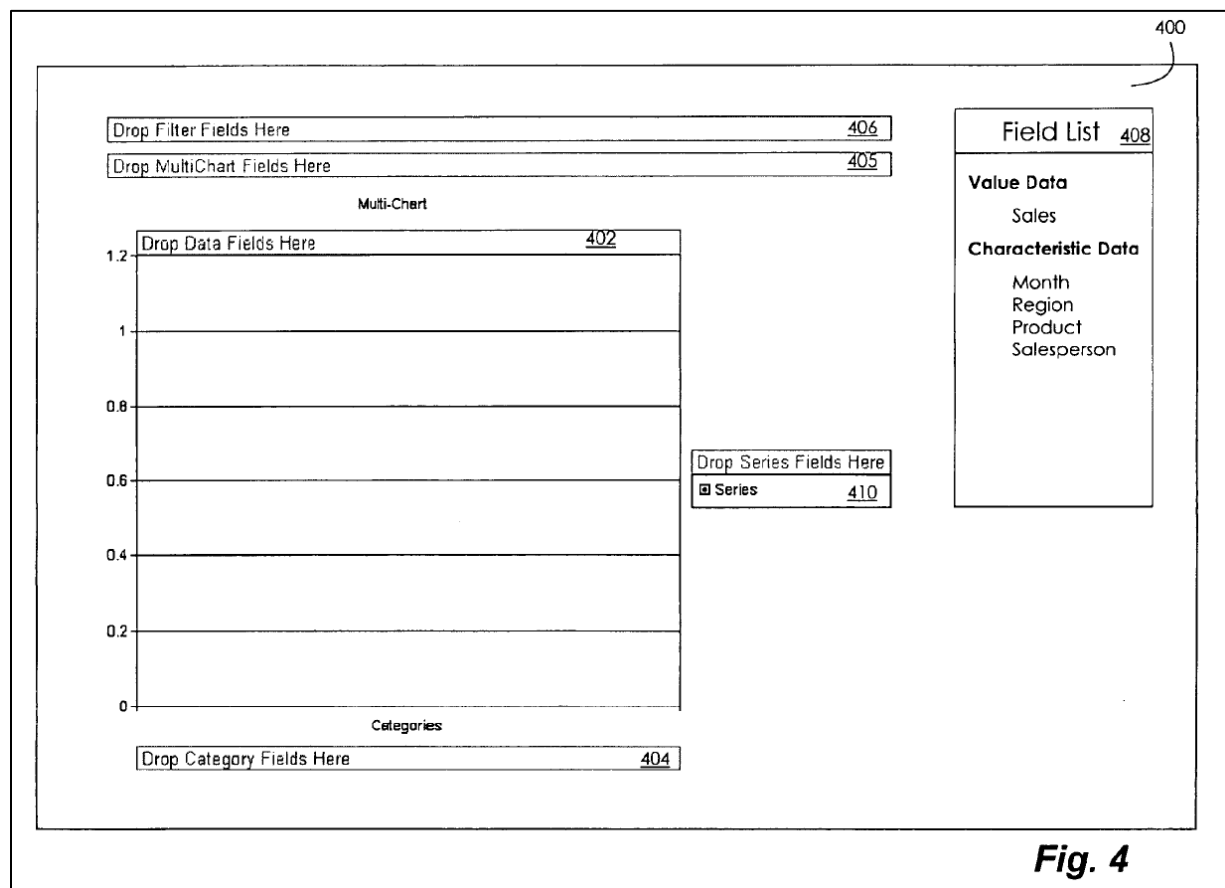
EX1006, FIG. 3B.

131. Figure 3C depicts a chart report generated from data source 300 at Figure 3A. EX1006, 6:20-22.



EX1006, FIG. 3C.

132. Figure 4 below shows Couckuyt's DDUI 400 for generating a chart. EX1006, 7:10-11. The DDUI may be associated with "data source 300" (shown above) causing "field list 408" to be generated. EX1006, 7:12-13. When Couckuyt's DDUI receives an indication that the user has dropped data into the chart pane (e.g., the user finishes a drag-and-drop operation by releasing the click of mouse 42), the DDUI automatically generates the interactive chart from the stored data. For example, "[d]ata [f]ield drop zone 402 can accept value data fields (e.g., sales) and will chart that data" EX1006, 7:24-28.



EX1006, FIG. 4.

133. Shown above, DDUI 400 includes category drop zone 404 and multi-chart drop zone 405. EX1006, 7:29-31. “If a user drops a characteristic data field in a Category drop Zone 404 that is already populated, then the newly dropped characteristic data field will replace the previously dropped characteristic data” EX1006, 7:43-47. In contrast, dropping additional data onto multi-chart drop zone 405 “will create at least one new chart.” EX1006, 7:51-53.

134. Figure 5, shown below, also demonstrates automatically generating a chart via drag and drop. DDUI 500 allows multiple data fields to be dropped, generating a sub-chart 502 for each sub-data field in the dropped data. EX1006, 7:67-8:4. For example, the “Education Level” data field may include two sub-data fields, “Bachelors Degree” and “Graduate Degree.” EX1006, 8:7-10. When “Education Level” is dragged and dropped onto the drop zone, the two sub-charts are generated. EX1006, 8:7-9.

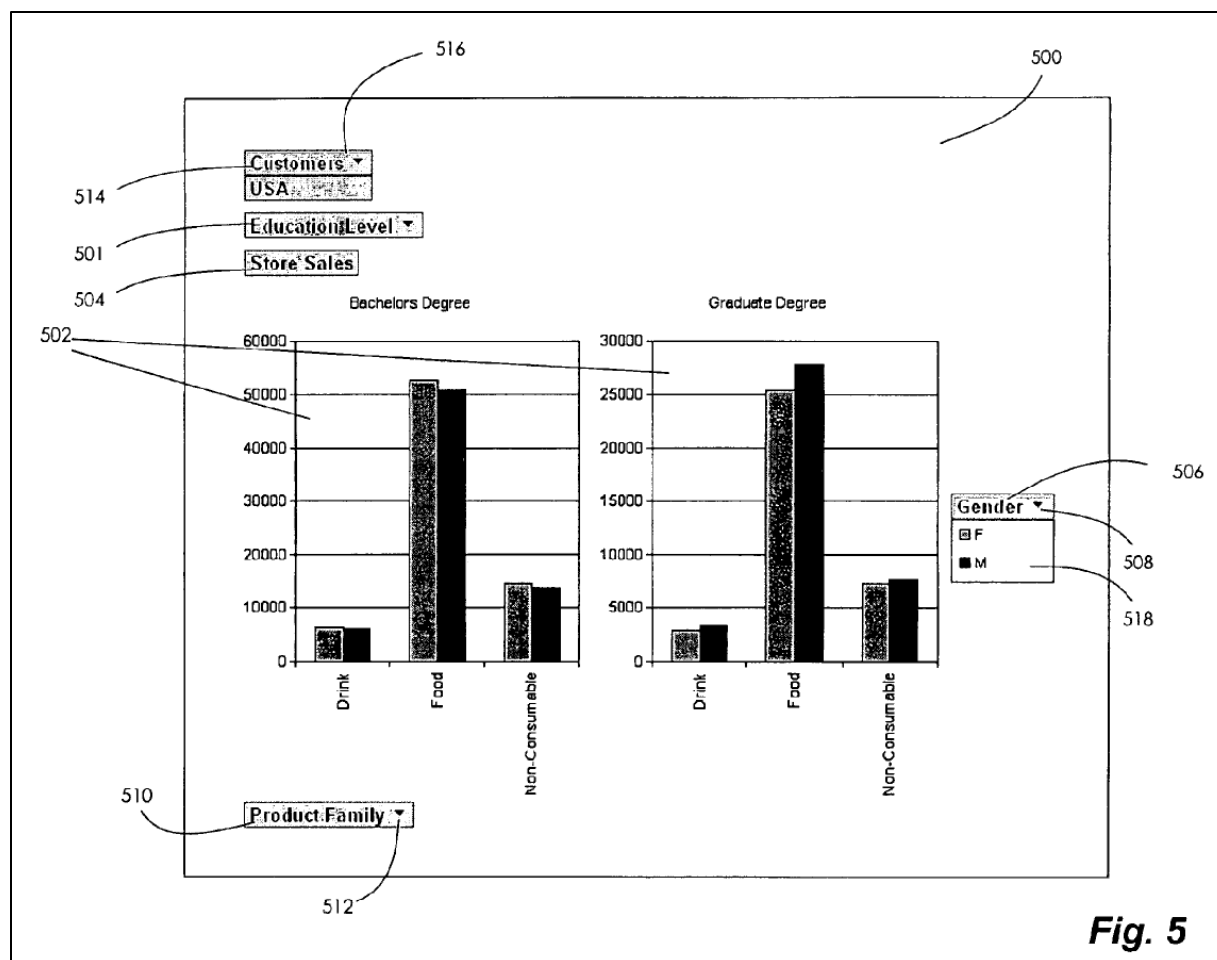


Fig. 5

EX1006, FIG. 5.

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

135. 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.

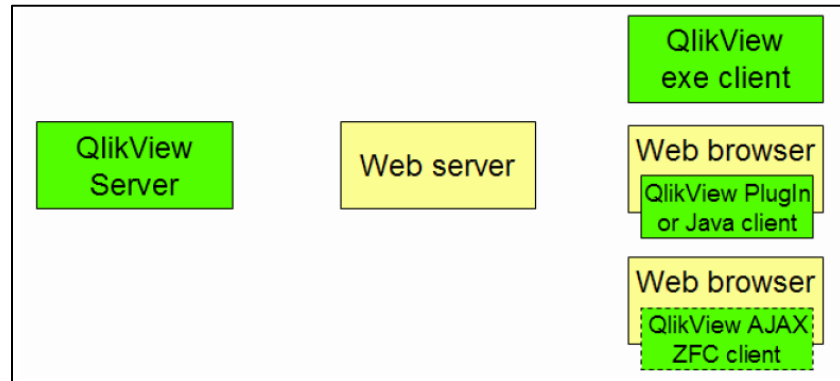
136. In early 2008, a POSA would have been aware of QlikView and the QlikView Manuals. The QlikView Manuals describe how QlikView works.

137. 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; EX1068, ¶7. 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.

138. 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.

139. 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.

140. 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.

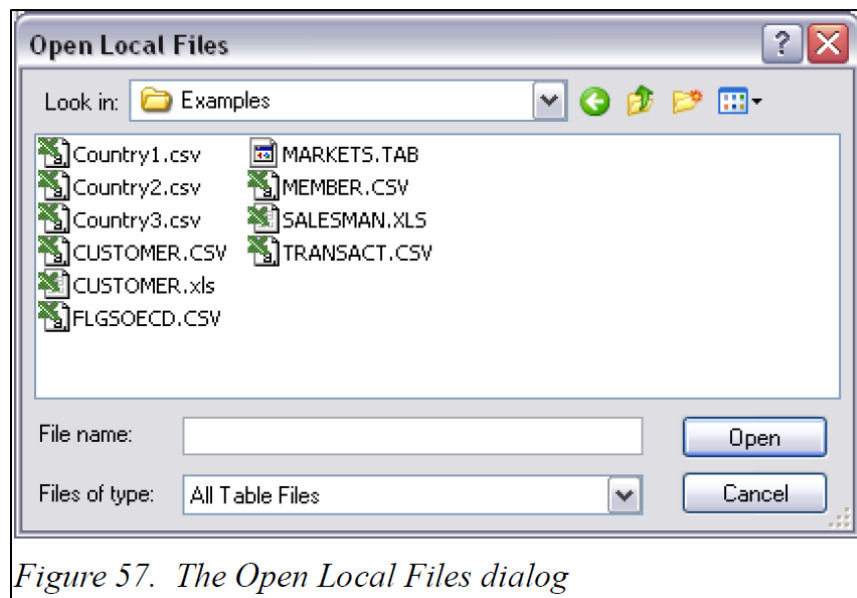


Figure 57. The Open Local Files dialog

EX1008, 0249.

141. 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.

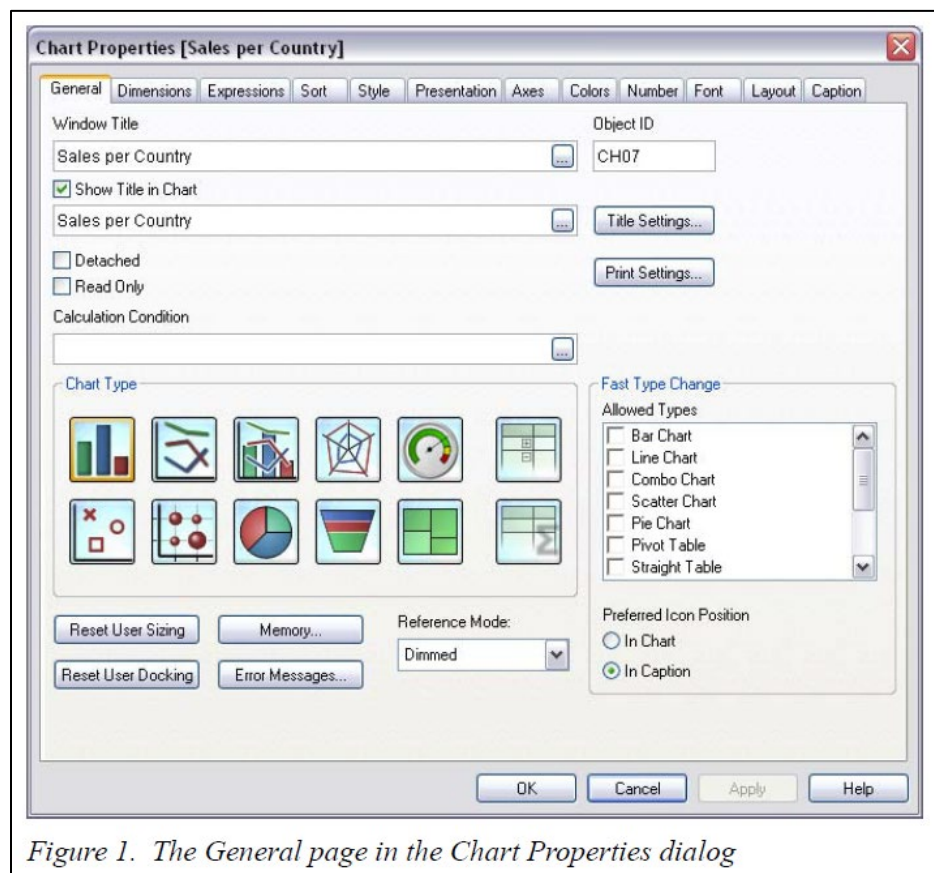
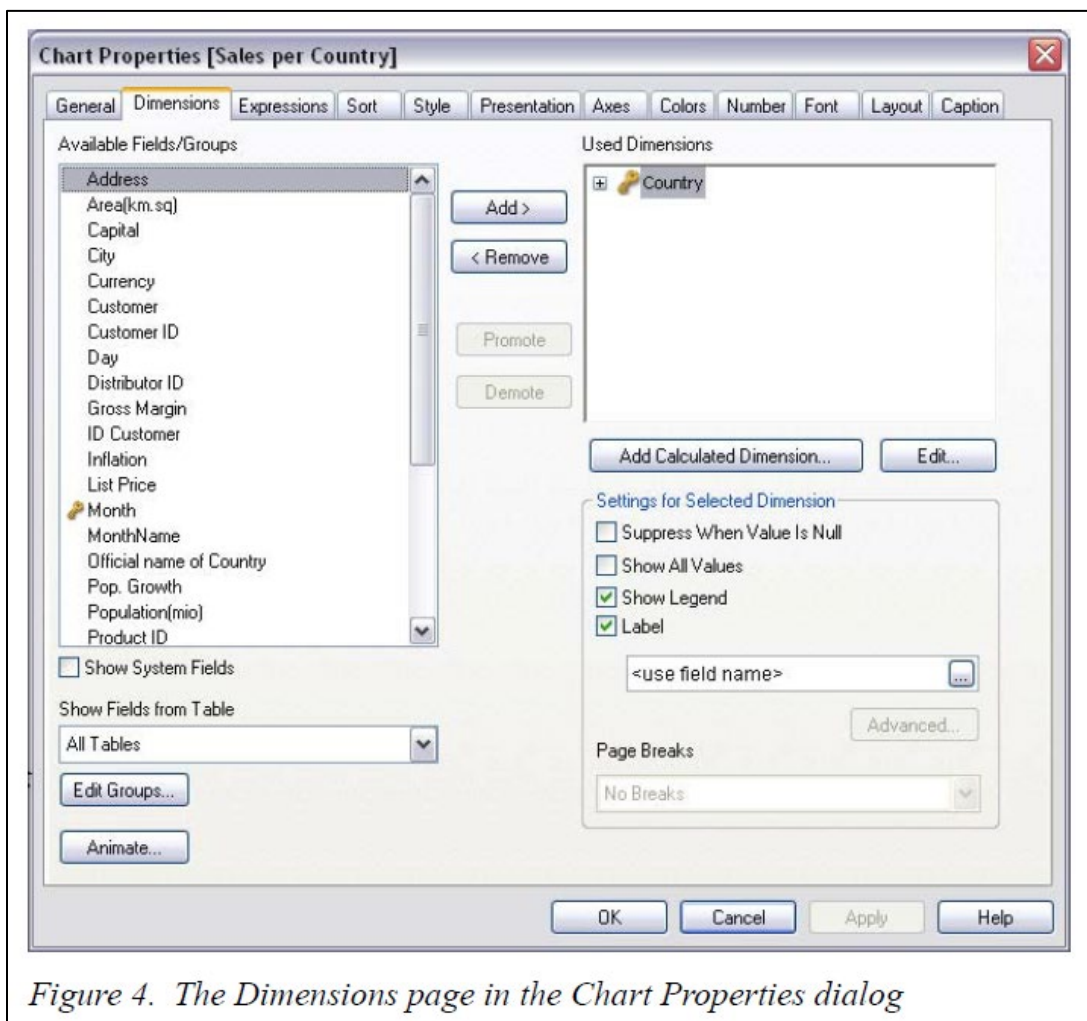


Figure 1. The General page in the Chart Properties dialog

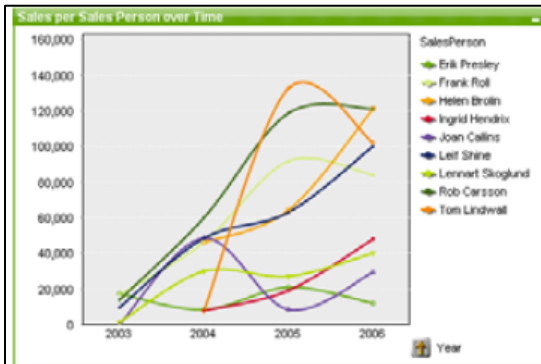
EX1008, 0882.

142. 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.

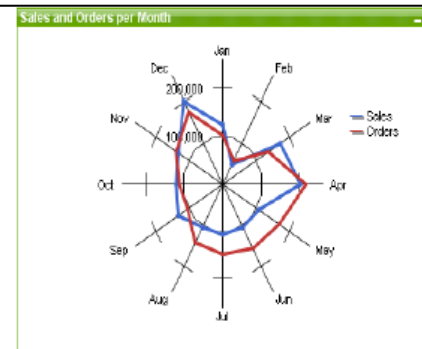


EX1008, 0894.

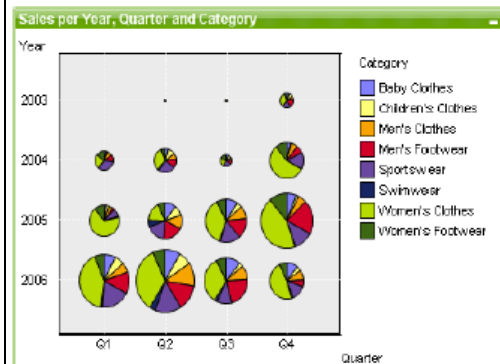
143. Below are example charts created by QlikView.



Line chart



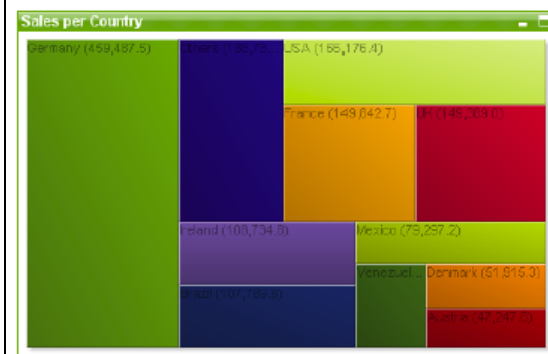
Radar chart



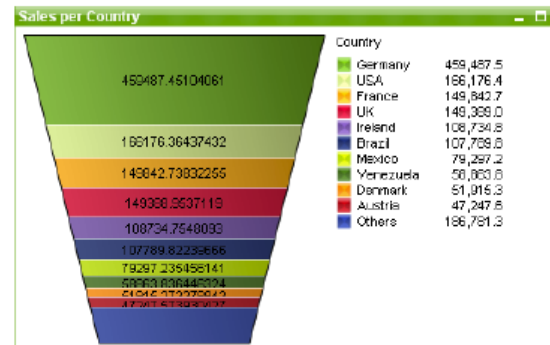
Grid chart



Gauge chart



Block chart

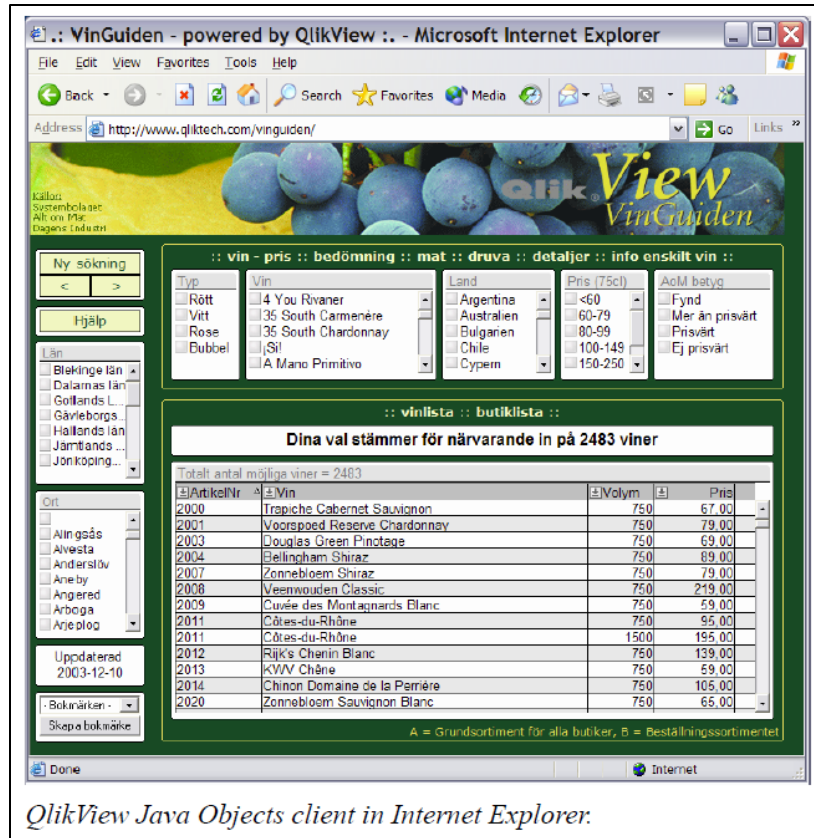


Funnel chart

EX1008, 0888.

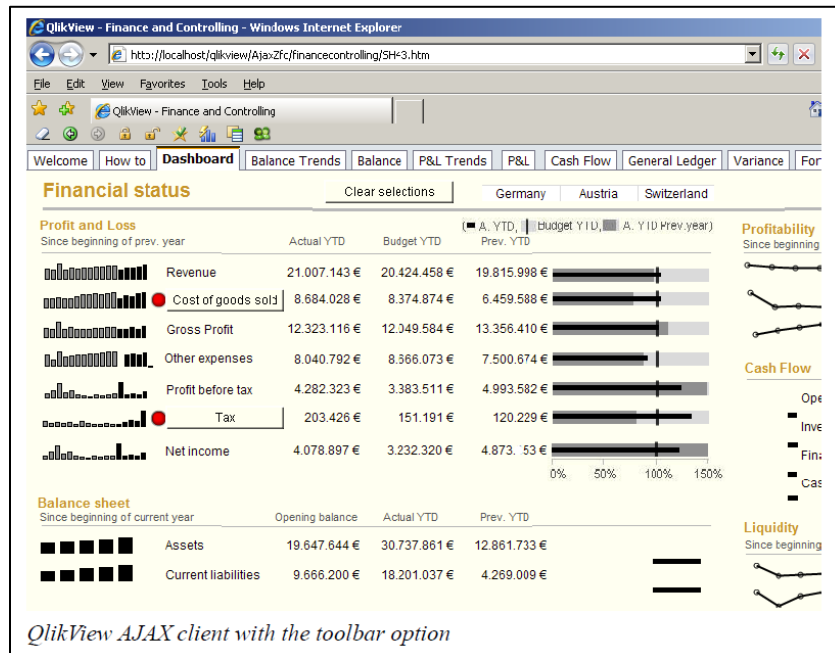
144. QlikView's Chart Wizard provided many customizable options, including interactive features such as "drop-down," "slider object[s]," "buttons," "interactive search," "drill-down" "cyclic groups," and "pop-up labels." EX1009, 0016-17; EX1008, 0035-36, 0583, 0919, 1306-07. For example, when customizing 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.

145. 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, 0153.

146. 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.

VIII. Grounds of Unpatentability

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

Ground	References	Claims
1	Rostoker	1, 2, 4-8, 10-14, 16-18
2	Rostoker, Clark	1, 2, 4-8, 10-14, 16-18
3	Rostoker, Jou	1, 2, 4-8, 10-14, 16-18
4	Rostoker, Clark, Jou	1, 2, 4-8, 10-14, 16-18
5	Rosotoker, Couckuyt	3, 9, 15
6	Rosotoker, Clark, Couckuyt	3, 9, 15
7	Rosotoker, Jou, Couckuyt	3, 9, 15
8	Rosotoker, Clark, Jou, Couckuyt	3, 9, 15
9	QlikView Manuals	1, 2, 4-8, 10-14, 16-18
10	QliKView Manuals, Couckuyt	3, 9, 15

IX. Ground 1: Rostoker renders obvious claims 1, 2, 4-8, 10-14, and 16-18.

148. Rostoker teaches or suggests every element in claims 1, 2, 4-8, 10-14, and 16-18.

A. Independent Claim 1

149. Rostoker teaches or suggests every element in claim 1.

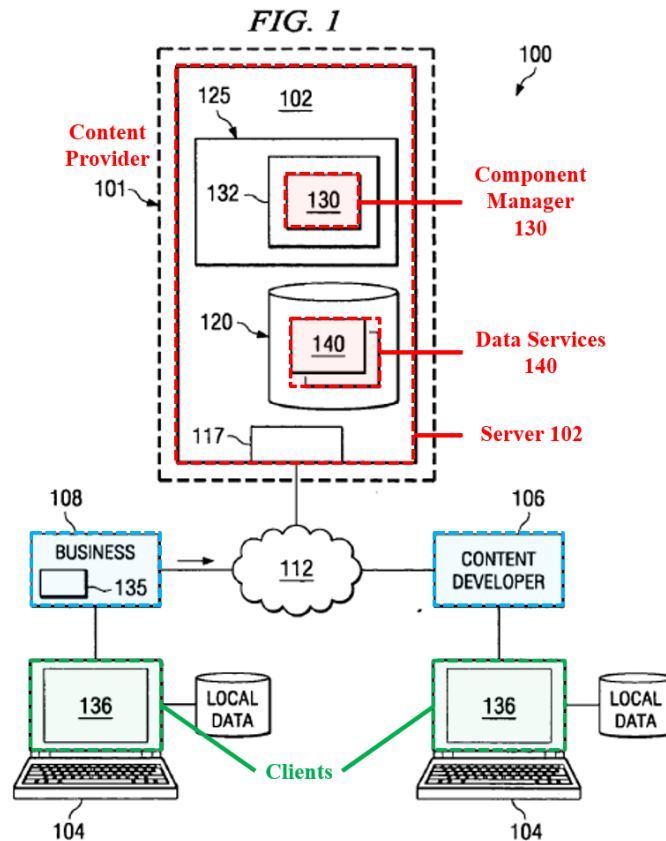
1. [1P] “A system for creating interactive charts”

150. The preamble of claim 1 recites “[a] *system for creating interactive charts.*” 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.

151. 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. “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.

152. 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.



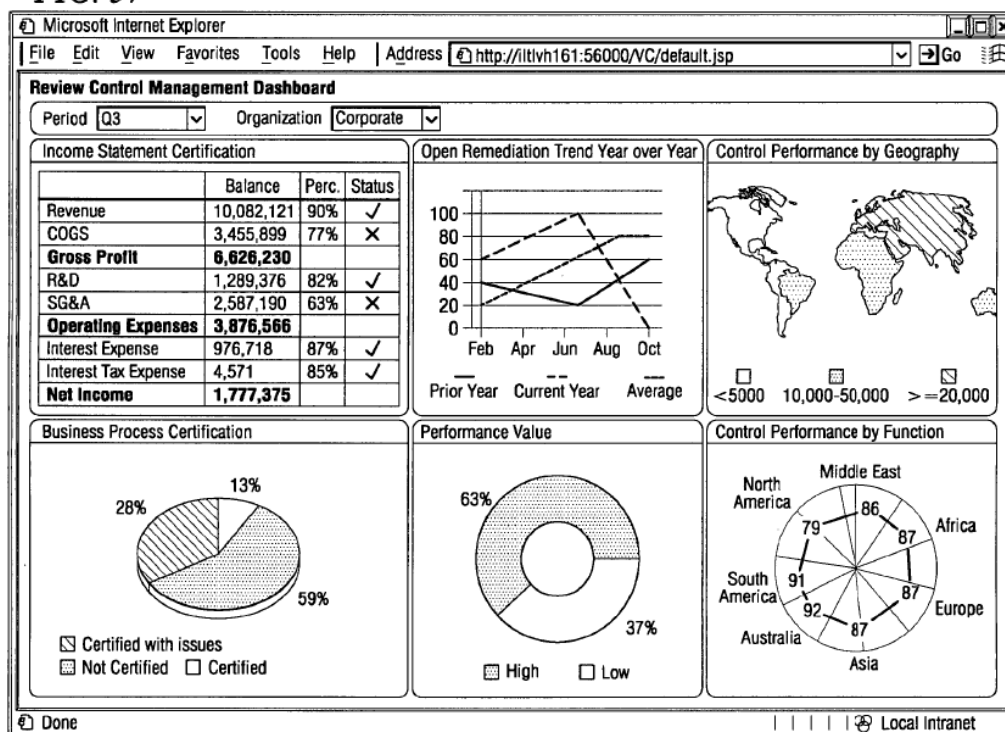
EX1005, FIG. 1 (annotated).

153. Rostoker discloses that “content provider 101 may be considered a storefront, hub, or retailer of business content and solutions, generally considered analytics implemented by business process components.” EX1005, ¶18. The “content provider 101 uses a component manager 130 . . . 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. Rostoker discloses that “[t]he 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. For example, “[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.

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

FIG. 57



EX1005, FIG. 57.

155. Additionally, as I discuss for element [1F] below, Rostoker teaches that its generated charts are “*interactive charts.*” See Section IX.B.7. Therefore, it is my opinion that Rostoker teaches the preamble of claim 1.

2. [1A] “memory that stores a chart template.

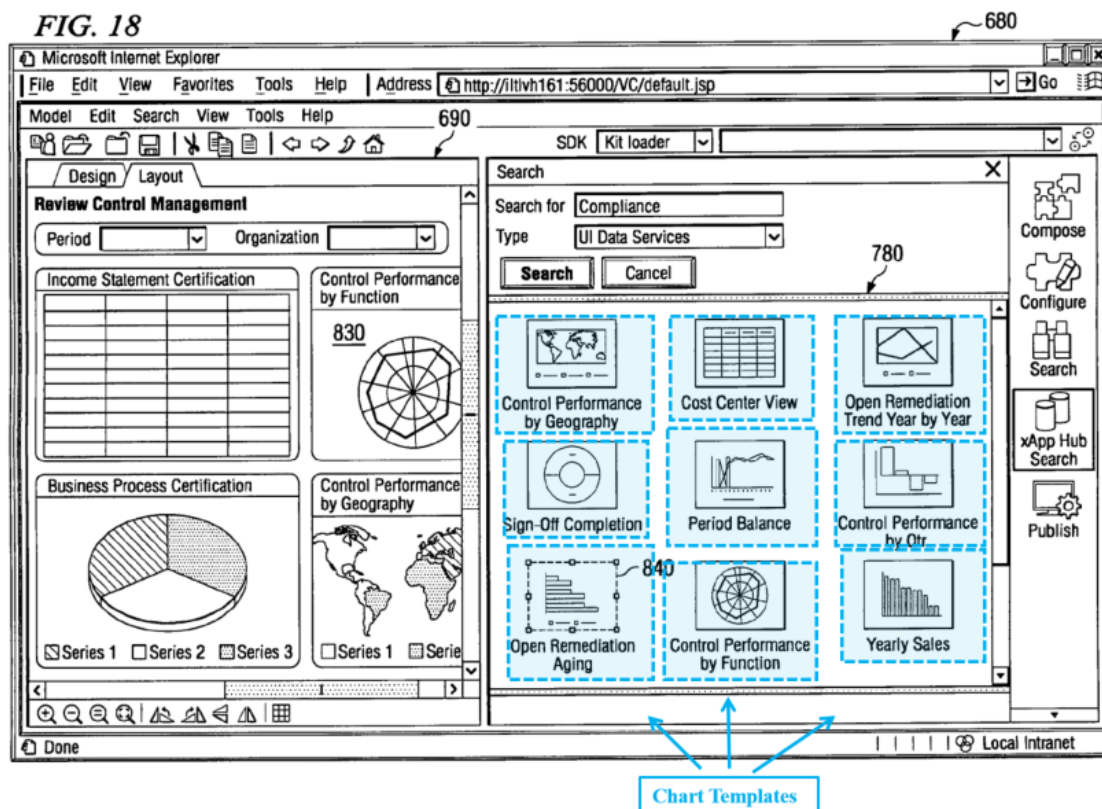
156. Element [1A] recites “*memory that stores a chart template.*” In my opinion, Rostoker teaches this limitation.

157. Rostoker discloses that its system 100 “is typically a distributed client/server system,” and server 102 includes “local memory 120.” EX1005, ¶¶28, 30, FIG. 1. Rostoker explains, “memory 120 includes data services 140” and “any other appropriate data.” EX1005, ¶30. Data service 140 “may include . . .

composite applications, analytics, **reusable components**, [and] objects”

EX1005, ¶¶30, 31. As I discuss for element [1P], Rostoker refers to these “composite applications”/“reusable components” as “**xApps**”/“**xApp components**.” EX1005, ¶4. For example, Rostoker describes, “[t]he present disclosure relates to methods, systems, and software for providing a compilation of **composite applications (‘xApps’)** and **xApp components** that may be modified, reconfigured, and/or adapted” EX1005, ¶4; *see also* EX1005, ¶22 (referencing “**composite applications 135 (or xApps)**”).

158. Rostoker also 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; *see also* EX1005, ¶21 (“[T]he content provider 101 may provide or utilize **a bank of easily searched xApps and xApp components (services/queries with prebuilt UIs) that can be recomposed/adapted to customer**, such as illustrated business 108, needs.”). An example of Rostoker’s “storefront”/“hub” with xApp templates is shown in Figure 18 below.



EX1005, FIG. 18 (annotated).

159. It is my opinion that a POSA would have understood that these reusable components or applications (e.g., xApps illustrated in Figure 18) are “*chart template[s]*” because, as Rostoker describes, they can be used “**as a template**, making modifications in order to create an application to operate in a desired manner.” EX1005, ¶6. And, as pictured in Figure 18, the available templates include bar charts, maps, line graphs, and donut charts. EX1005, FIG. 18.

160. A POSA also would have understood that these “*chart template[s]*” are stored in Rostoker’s “memory 120.” Rostoker explains that “Memory 120 may

include any memory or database module and may take the form of volatile or non-volatile memory including, without limitation, magnetic media, optical media, random access memory ('RAM'), read-only memory ('ROM'), removable media, or any other suitable local or remote memory component.” EX1005, ¶30. Rostoker further explains that “memory 120 includes data services 140,” and “data services 140 may include (among other things) **composite applications**, analytics, [and] **reusable components**” EX1005, ¶¶30, 31. In other words, memory 120 stores the reusable applications and components available through Rostoker’s “storefront”/“hub.” EX1005, ¶¶6, 21, 30, 31. Therefore, Rostoker’s memory 120 provides “*memory that stores a chart template.*”

161. Therefore, it is my opinion that Rostoker teaches element [1A].

3. [1B] “**a processor, wherein the processor executes instructions stored in the memory, causing the processor to:**”

162. Element [1B] recites “*a processor, wherein the processor executes instructions stored in the memory, causing the processor to [perform certain steps].*” In my opinion, Rostoker teaches this limitation.

163. Rostoker discloses that server 102 includes “**processor 125**,” which “**executes instructions** and manipulates data to perform the operations of server 102.” EX1005, ¶33. Rostoker further explains that “processor 125 executes at least component manager 130,” which “is software operable to provide an interface to a

plurality of solutions and some or all of the underlying reusable components 140 that can be used for these solutions.” EX1005, ¶¶33-34.

164. As Rostoker explains, component manager 130 presents “a plurality of data services 140 to businesses 108, or their individual users,” and it “may form a portion of a development or modeling environment 132.” EX1005, ¶19; *see also* EX1005, ¶34 (“[C]omponent manager 130 is any application, program, module, process, or other software that helps manage at least a portion of data services 140 and is operable to present certain components 140 though interface 136 for viewing and potential use or purchase by others.”). As I discuss for element [1A], local memory 120 of server 102 “includes data services 140” and “any other appropriate data,” including “software applications or sub-systems.” EX1005, ¶30. Therefore, it is my opinion that a POSA would have understood that Rostoker teaches “a processor,” e.g., processor 125, that “*executes instructions stored in the memory*”—that is, instructions to execute Rostoker’s modeling environment, including component manager 130.

4. [1C] “generate a user interface on a first website”

165. Element [1C] recites a processor that “*generate[s] a user interface on a first website.*” In my opinion, Rostoker teaches this limitation.

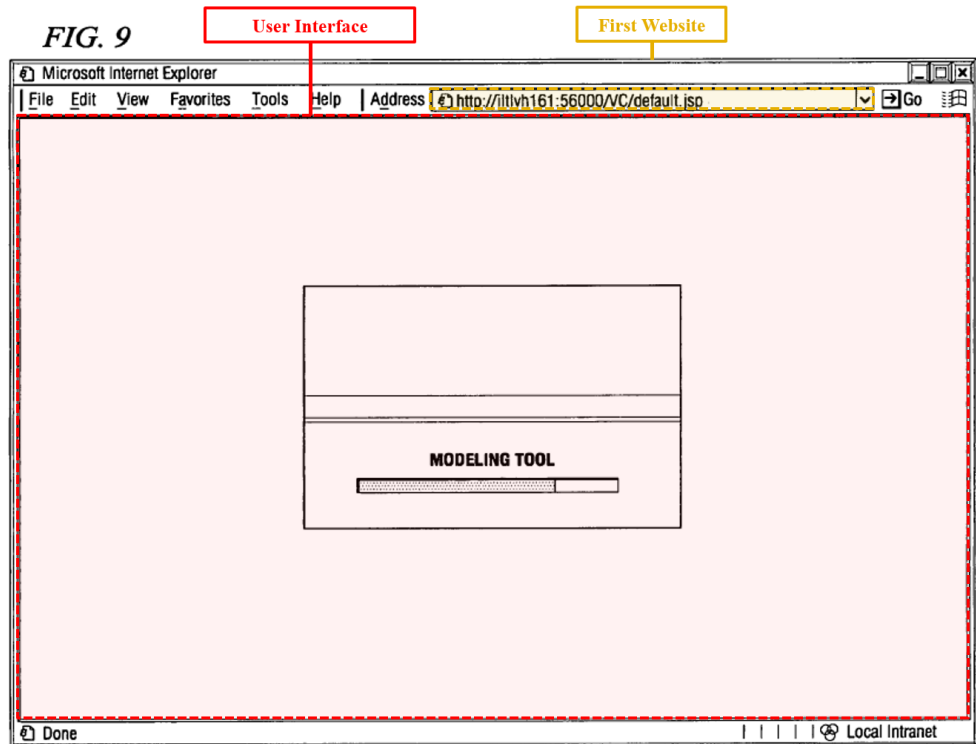
166. Rostoker discloses a “*user interface*” in the form of a “modeling environment 132.” EX1005, ¶40. For example, Rostoker discloses that “[t]he

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.”

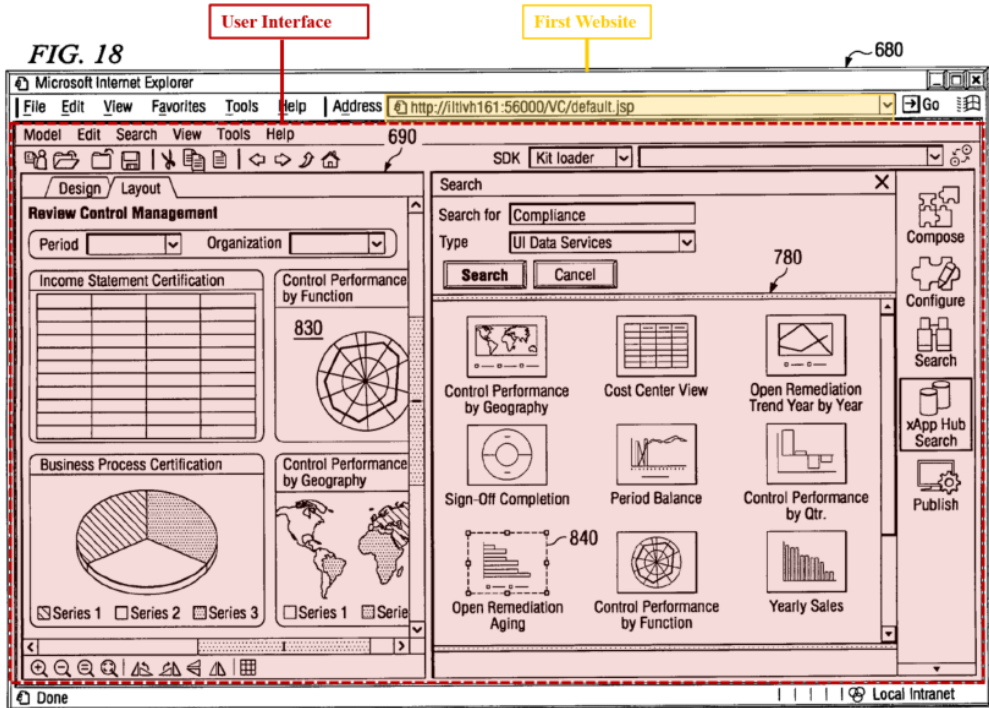
EX1005, ¶40.

167. Rostoker explains that “The component manager 130 may form a portion of a development or modeling environment 132.” EX1005, ¶19. A “user may access the modeling environment 132 via the component manager 130,” and “the 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.” EX1005, ¶¶28, 56. Rostoker also discloses that content provider 101 (which includes the component manager 130, as shown in Figure 1) “can typically be accessed in a stand-alone fashion (**such as a website**).” EX1005, ¶18; *see also* EX1005, ¶29 (“[S]erver 102 [of content provider 101] may also include or be communicably coupled with **a web server . . .**”).

168. Rostoker’s “modeling environment” is shown throughout Rostoker’s figures, for example in Figures 9 and 18. EX1005, ¶¶66-67. As illustrated in these figures, Rostoker’s user interface is generated and presented “*on a first website*.” EX1005, ¶¶66-67, FIGs. 9, 18.



EX1005, FIG. 9 (annotated).



EX1005, FIG. 18 (annotated).

169. For example, as illustrated in Figures 9 and 18, Rostoker's modeling environment is accessed from a web browser, in this case "Microsoft Internet Explorer." EX1005, FIGs. 9, 18. The "Address" bar (highlighted in yellow in the figures above) shows the address of the website in which the user interface is generated and presented to the user. EX1005, FIGs. 9, 18.

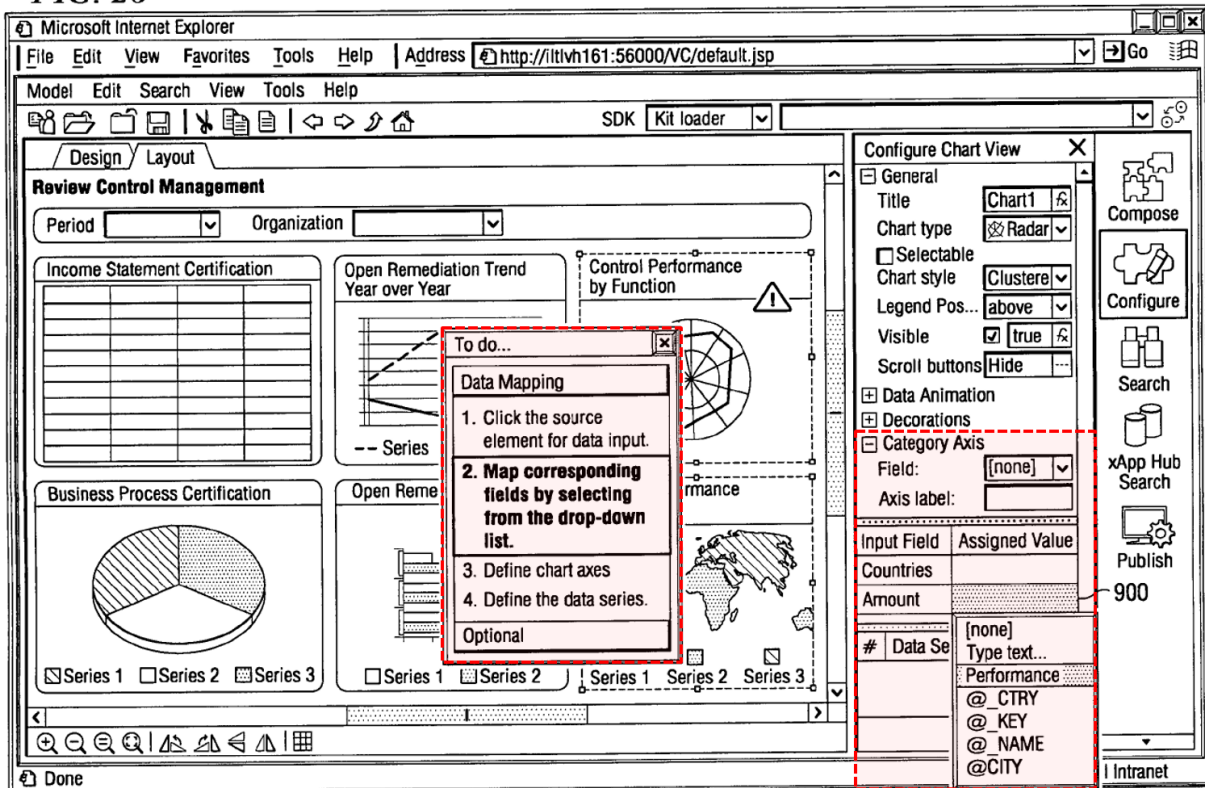
170. Therefore, it is my opinion that Rostoker teaches "*generat[ing] a user interface on a first website.*"

5. [1D] "receive chart data via the user interface"

171. Element [1D] recites a processor that "*receive[s] chart data via the user interface.*" In my opinion, Rostoker teaches this limitation.

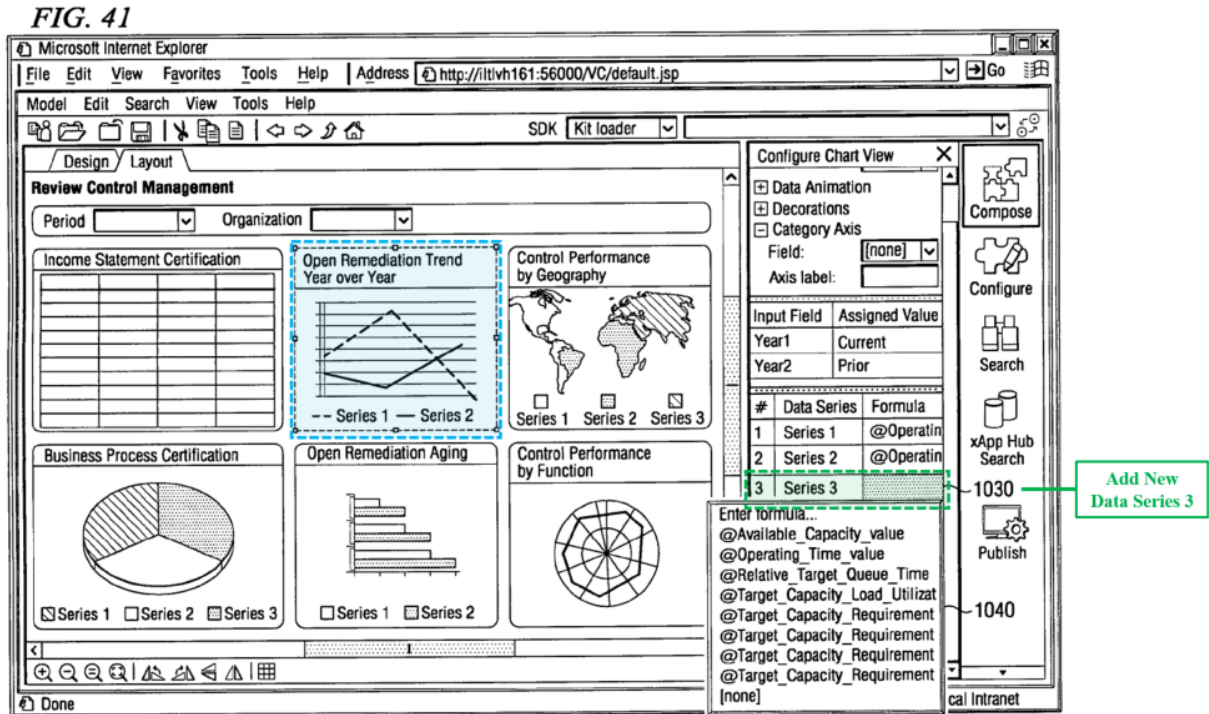
172. Rostoker discloses a "Data Mapping" function in its user interface to map data to a chart, and specific data fields may be mapped to chart axes, etc., as illustrated in Figure 26 of Rostoker below. EX1005, ¶68, FIG. 26.

FIG. 26



EX1005, FIG. 26 (annotated).

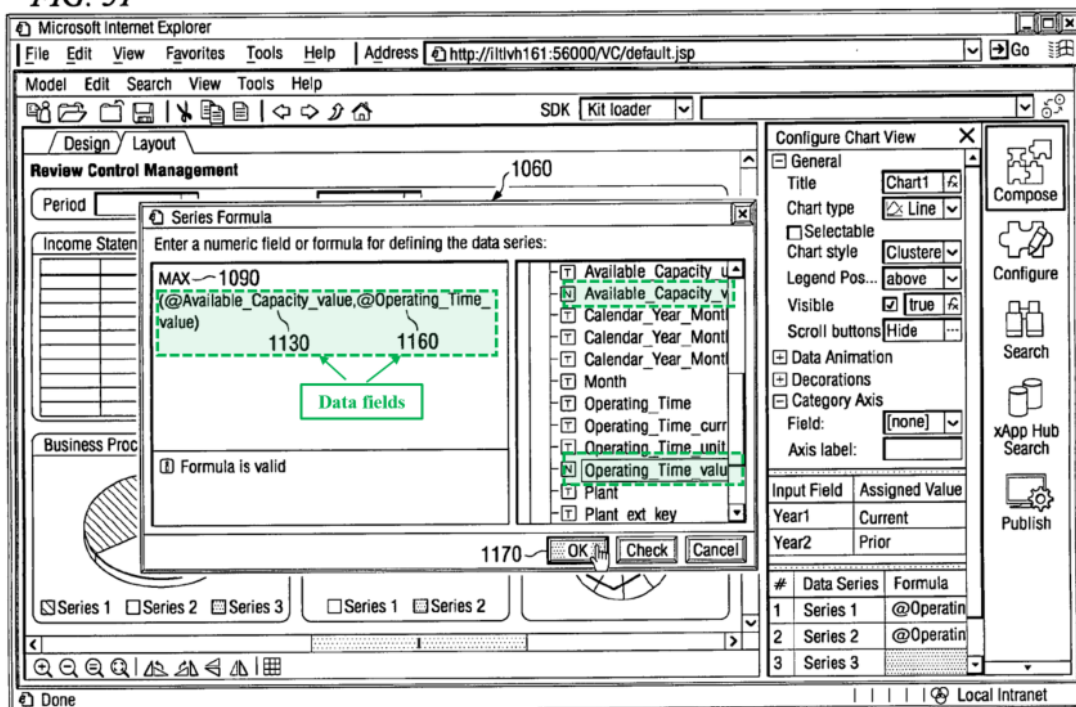
173. As part of its “Data Mapping” function, Rostoker discloses a function for adding “a new data series” to a chart in its user interface. EX1005, ¶70. As an example, Figure 41 of Rostoker below illustrates that the chart “Open Remediation Trend Year over Year” includes Data Series 1 and Data Series 2. EX1005, FIG. 41.



EX1005, FIG. 41 (annotated).

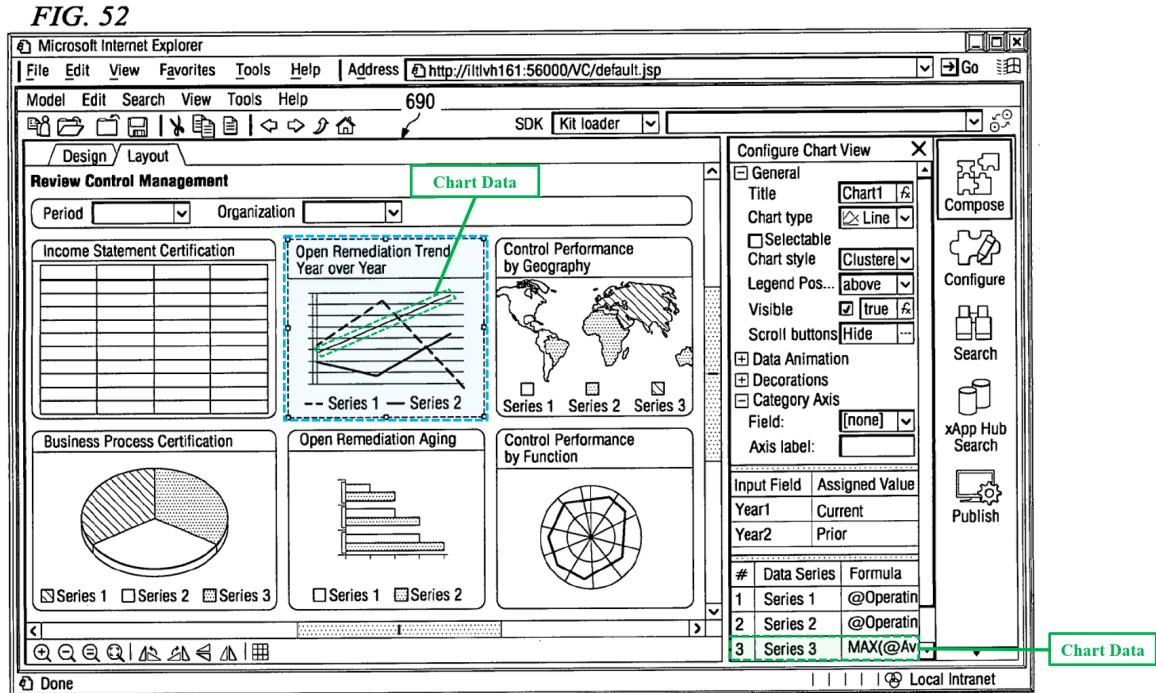
174. The interface in Figure 41 also indicates “the addition of a new data series 1030 and a corresponding list 1040 of available data series for the new data series 1030.” EX1005, ¶70, FIG. 41. Figure 51 of Rostoker below shows that data field 1130 (“Available_Capacity_value”) and data field 1160 (“Operating_time_value”) are selected as variables of a mathematical function 1090 for the new Data Series 3. EX1005, ¶70, FIG. 51. The user is then able to map the data to the chart by clicking “OK button 1170.” EX1005, ¶70.

FIG. 51



EX1005, FIG. 51 (annotated).

175. Once the data is received through Rostoker's user interface, for example by clicking "OK button 1170," the new data is mapped to the chart, as shown below in Figure 52. EX1005, ¶70 ("FIG. 52 shows that the window 1060 has been removed and the UI 690 of the application is again displayed.").



EX1005, FIG. 52 (annotated).

176. Therefore, it is my opinion that Rostoker discloses “*receiv[ing] chart data via the user interface.*”

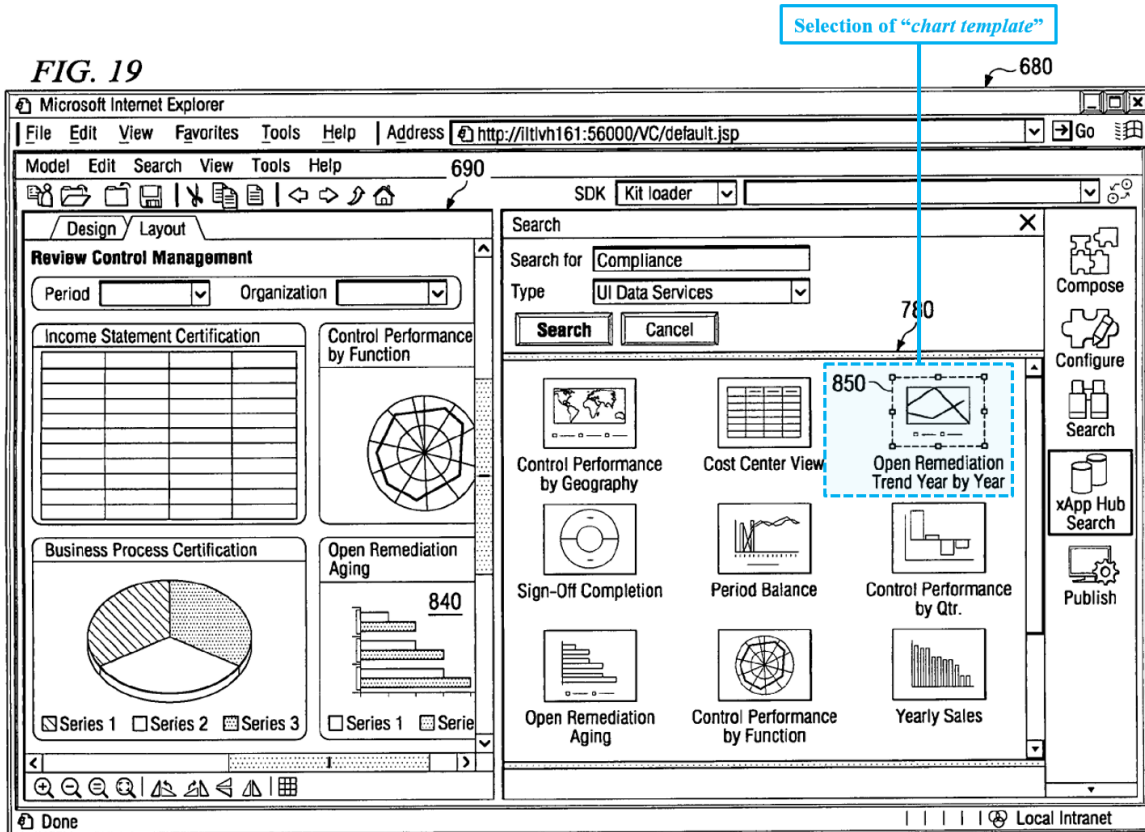
6. [1E] “**receive a selection of the chart template via the user interface**”

177. Element [1E] recites a processor that “*receive[s] a selection of the chart template via the user interface.*” In my opinion, Rostoker teaches this limitation.

178. As I discuss for element [1A], Rostoker discloses “*chart template[s].*” See Section IX.B.2. For example, through 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, 21 (“[T]he content provider 101 may provide or utilize a bank of easily searched xApps and xApp components (services/queries with prebuilt UIs) that can be recomposed/adapted to customer, such as illustrated business 108, needs.”).

179. Rostoker explains that the user “may locate the components by searching the hub.” EX1005, ¶¶6, 21; *see also* EX1005, ¶59 (“The user may also identify pre-selected objects by performing additional searches of the hub.”). Figure 19 of Rostoker below illustrates selection of “*the chart template*,” in this case xApp 850 (“Open Remediation Trend Year by Year”), within Rostoker’s development user interface. EX1005, ¶67 (“**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.**”).



EX1005, FIG. 19 (annotated).

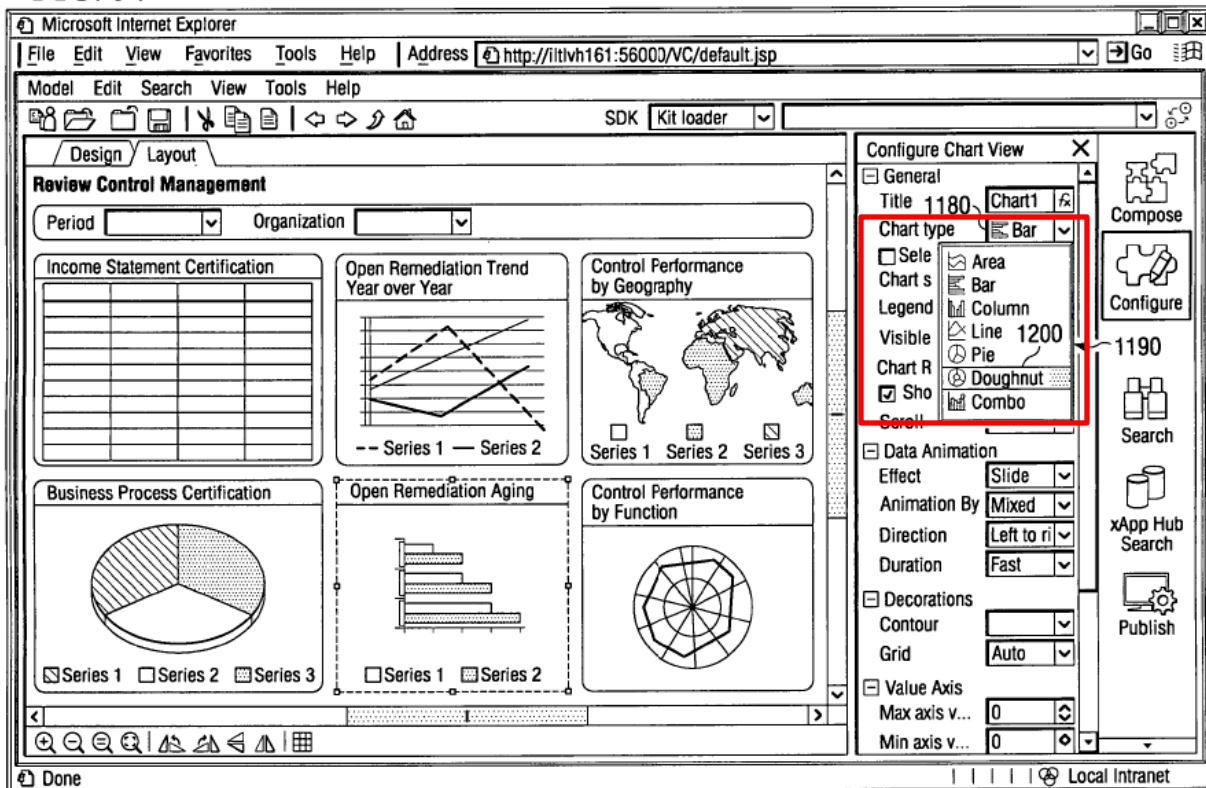
180. Rostoker explains that such a selection can be made using “an input device, such as a keypad, touch screen, mouse, or other device that can accept information.” EX1005, ¶38. Following selection, the “chart template” is added to the current application, as illustrated in Figure 20 of Rostoker below. EX1005, ¶68 (“FIG. 20 shows the xApps included in the application's UI 690,” such as “xApps 820-850”), FIG. 20.



EX1005, FIG. 20 (annotated).

Rostoker teaches element [1E].

FIG. 54



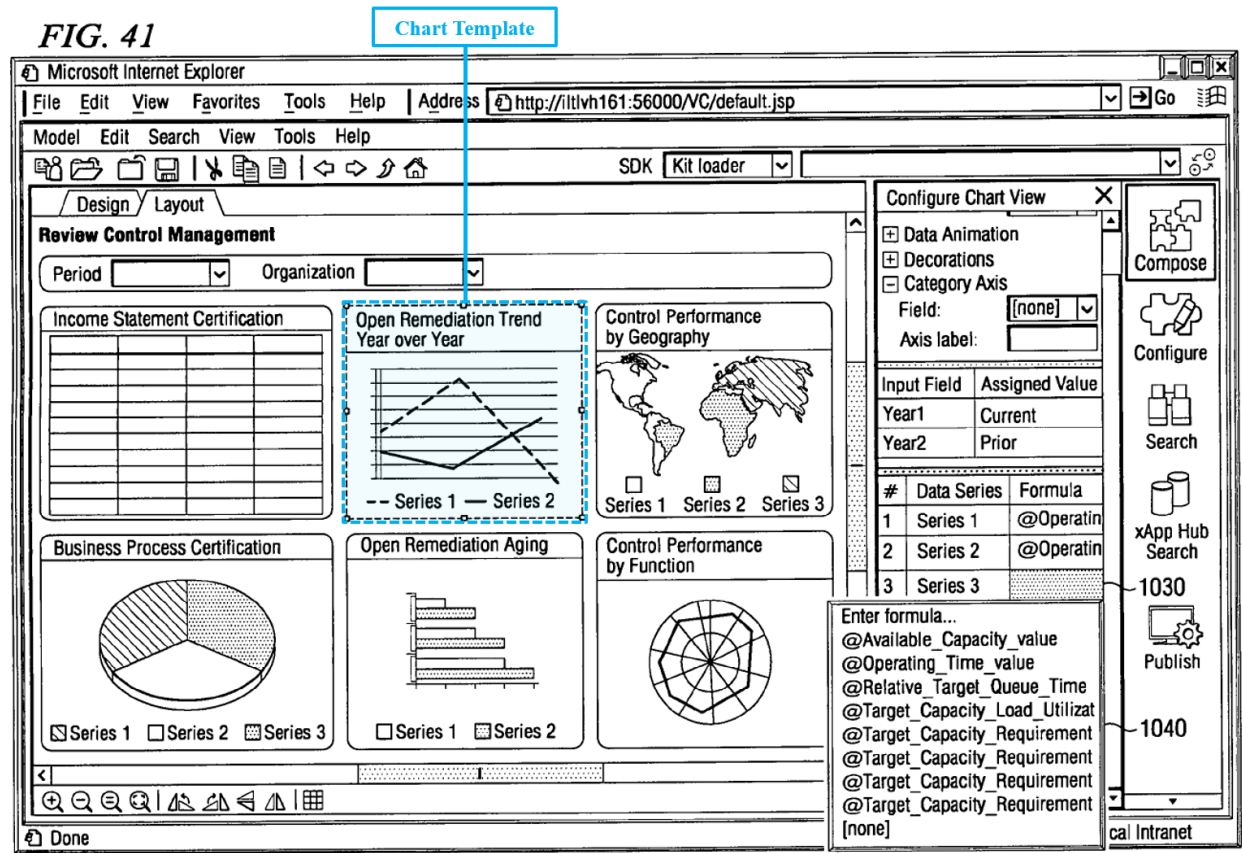
EX1005, FIG. 54 (annotated).

7. [1F] “generate an interactive chart including the chart data and the chart template”

182. Element [1F] recites a processor that “*generate[s] an interactive chart including the chart data and the chart template.*” In my opinion, Rostoker teaches this limitation.

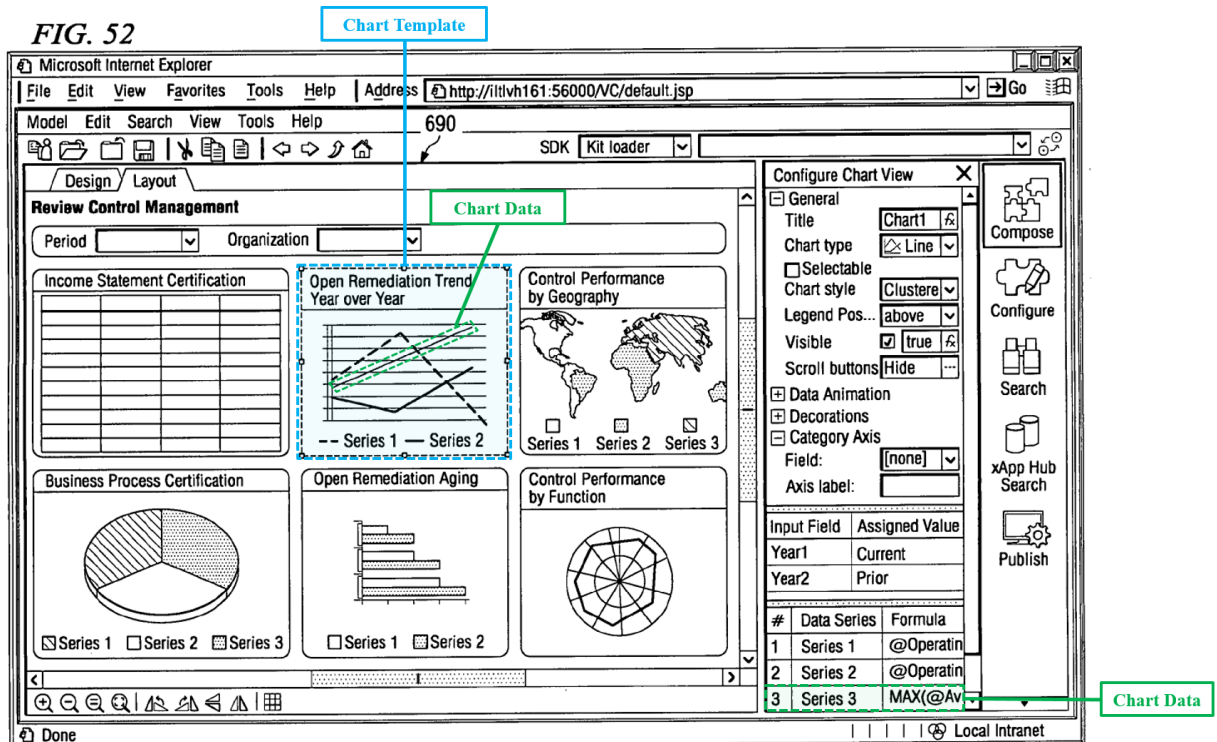
183. As I discuss for elements [1D] and [1E], Rostoker discloses selecting a “*chart template*” (e.g., xApp 850) and “*chart data*” using Rostoker’s user interface. See Sections IX.B.5, IX.B.6. For example, Figure 41 of Rostoker below

illustrates the “Open Remediation Trend Year over Year” “*chart template*” after selection.



EX1005, FIG. 41 (annotated).

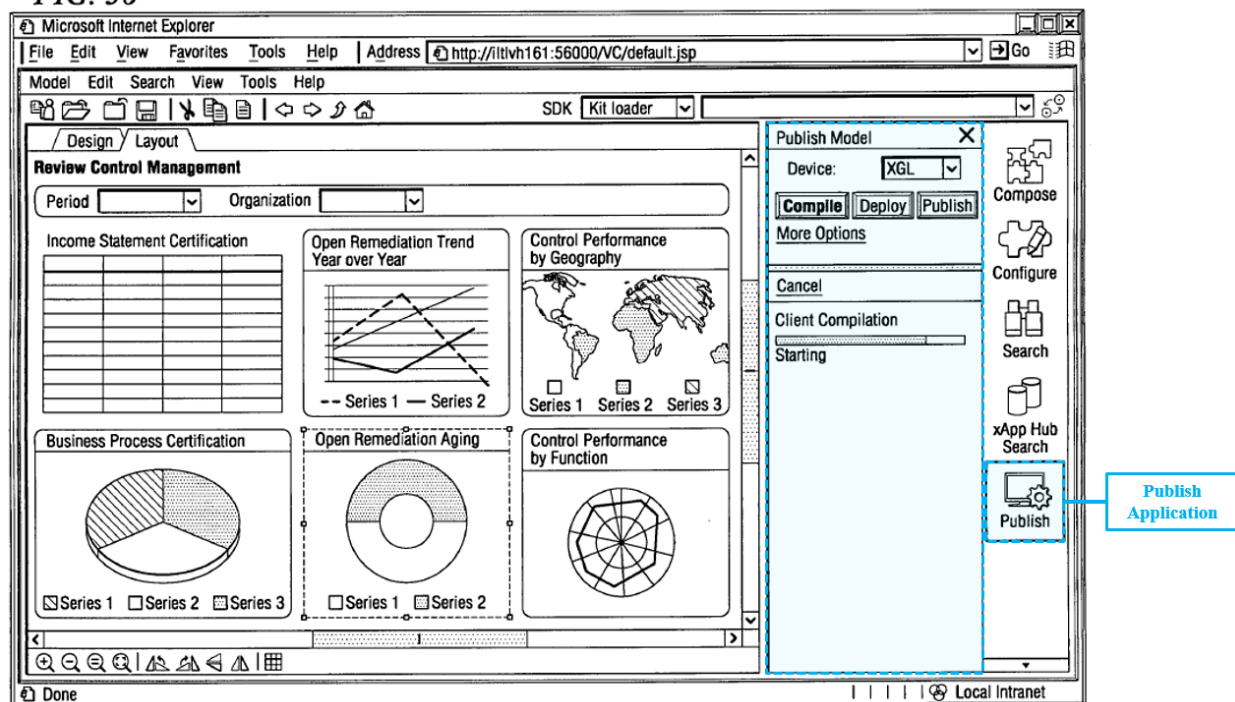
184. As I discuss for element [1D], after “the addition of a new data series,” the chart is updated to include the new data. See Section IX.B.5; EX1005, ¶70, FIG. 52. Figure 52 of Rostoker below illustrates the chart including the new data series. EX1005, ¶70, FIG. 52.



EX1005, FIG. 52 (annotated).

185. Rostoker then teaches “*generat[ing]*” a “*chart including the chart data and the chart template*” when the chart is published. In particular, Rostoker discloses a “Publish” function that allows a user to publish and deploy an application and charts included in the application. EX1005, ¶71, FIG. 56. As shown in Figure 56 of Rostoker below, as part of publishing the application and included charts, the application components are “transformed to XGL [(‘executable GUI language’)] representations.” EX1005, ¶42, FIG. 56 (showing “XGL” in the “Publish Model” toolbar).

FIG. 56

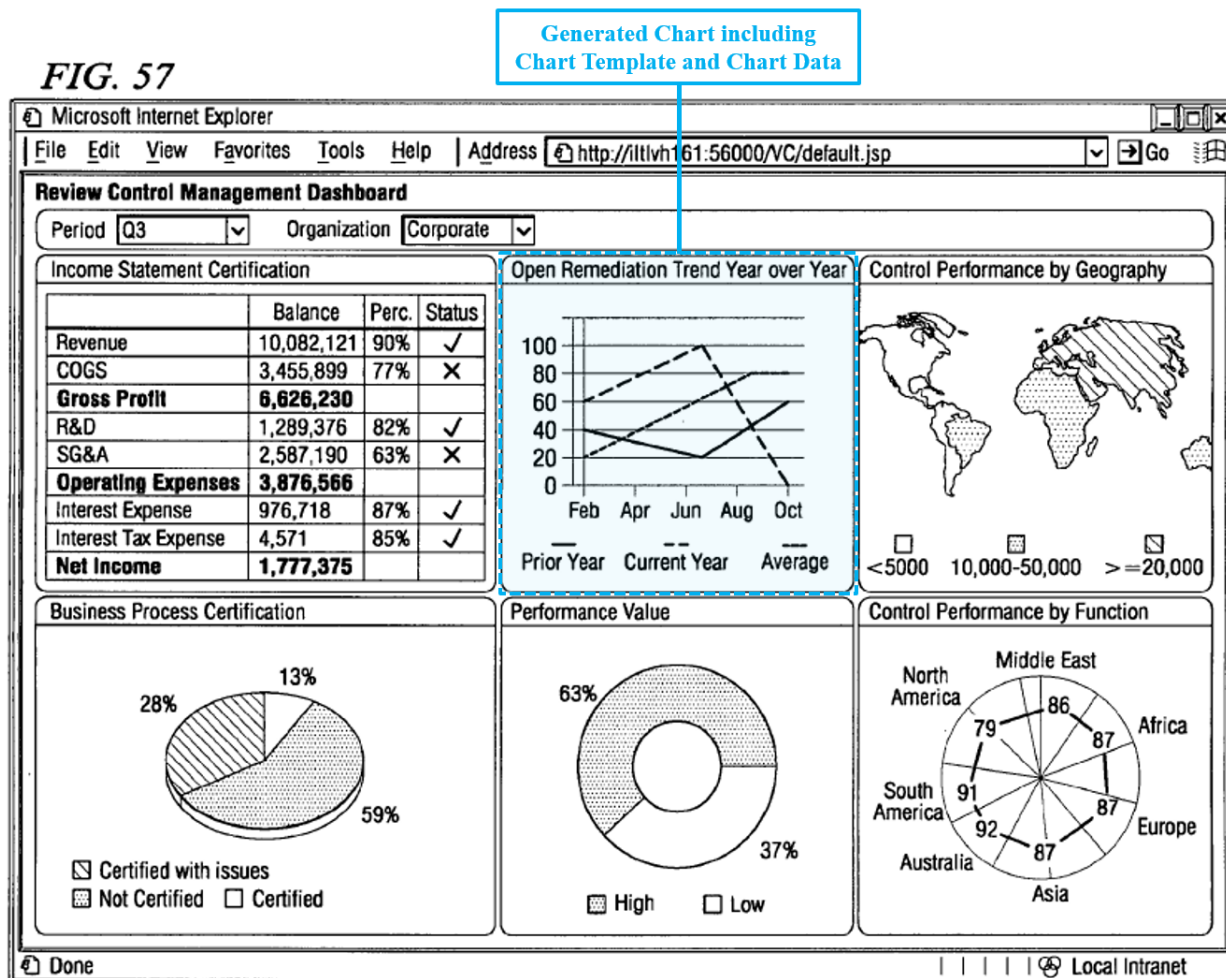


EX1005, FIG. 56 (annotated).

186. 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 then 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. Based on these disclosures, a POSA would have understood that, as part of Rostoker’s publishing function, Rostoker teaches

generating Java or Flash representations of applications and their included charts for execution in a “Java runtime” or “Flash runtime.” EX1005, ¶50. Indeed, it was well known that Flash, for example, enables the type of “rich data visualizations and animations, rich modes of data entry and user interactions,” EX1005, ¶43, that Rostoker discusses. *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.”).

187. Once published, “FIG. 57 shows the new or modified application after execution,” which includes the generated chart. EX1005, ¶71.

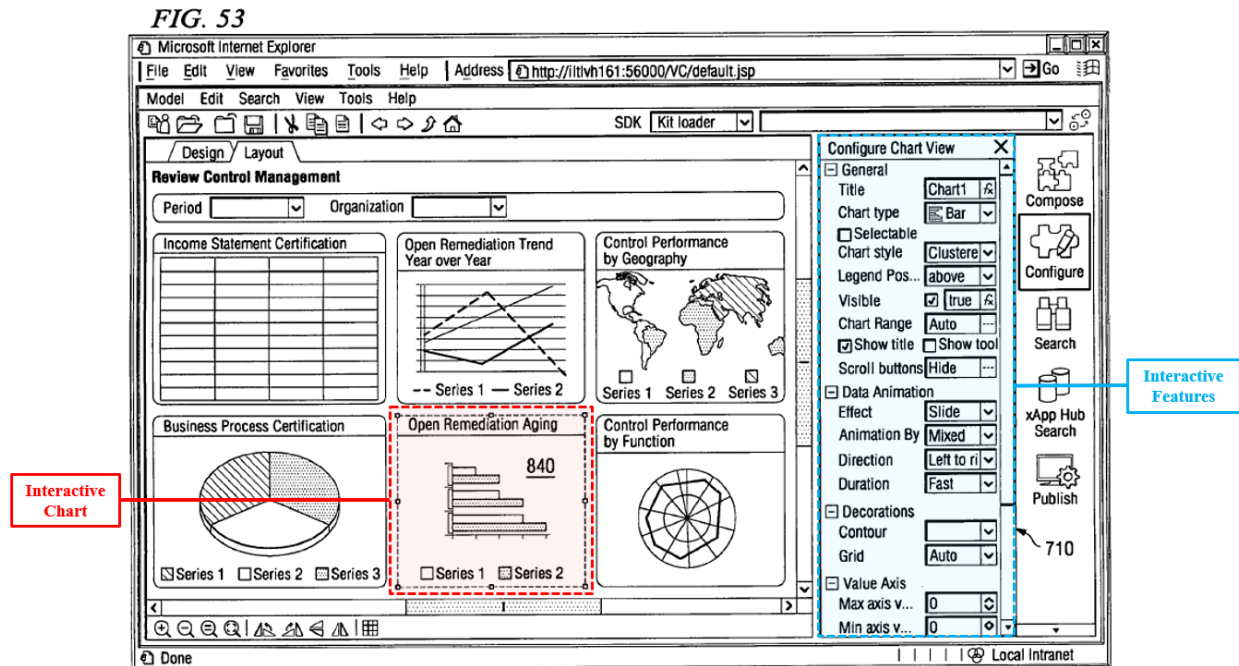


EX1005, FIG. 57 (annotated).

188. Therefore, it is my opinion that Rostoker discloses “*generat[ing]*” a “*chart including the chart data and [] chart template.*”

189. Rostoker also discloses that the generated chart is “*an interactive chart*” for multiple reasons. 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.

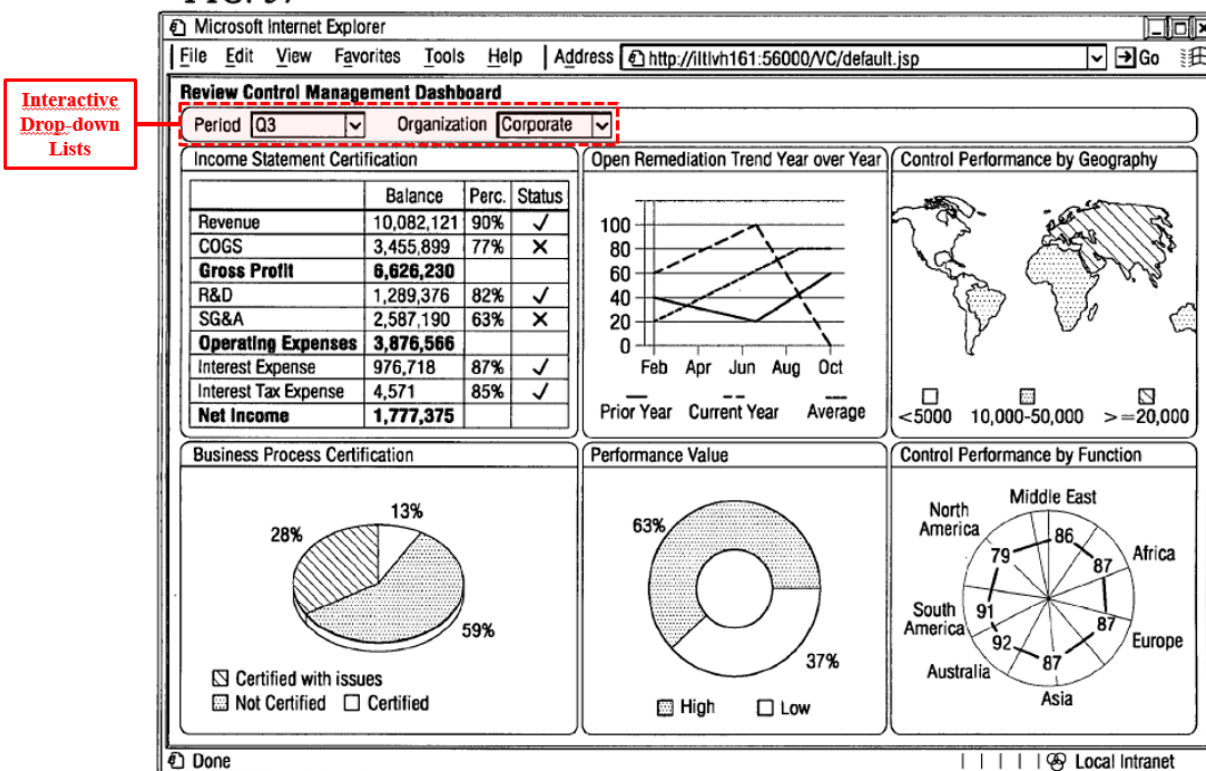


EX1005, FIG. 53 (annotated).

190. 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 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.

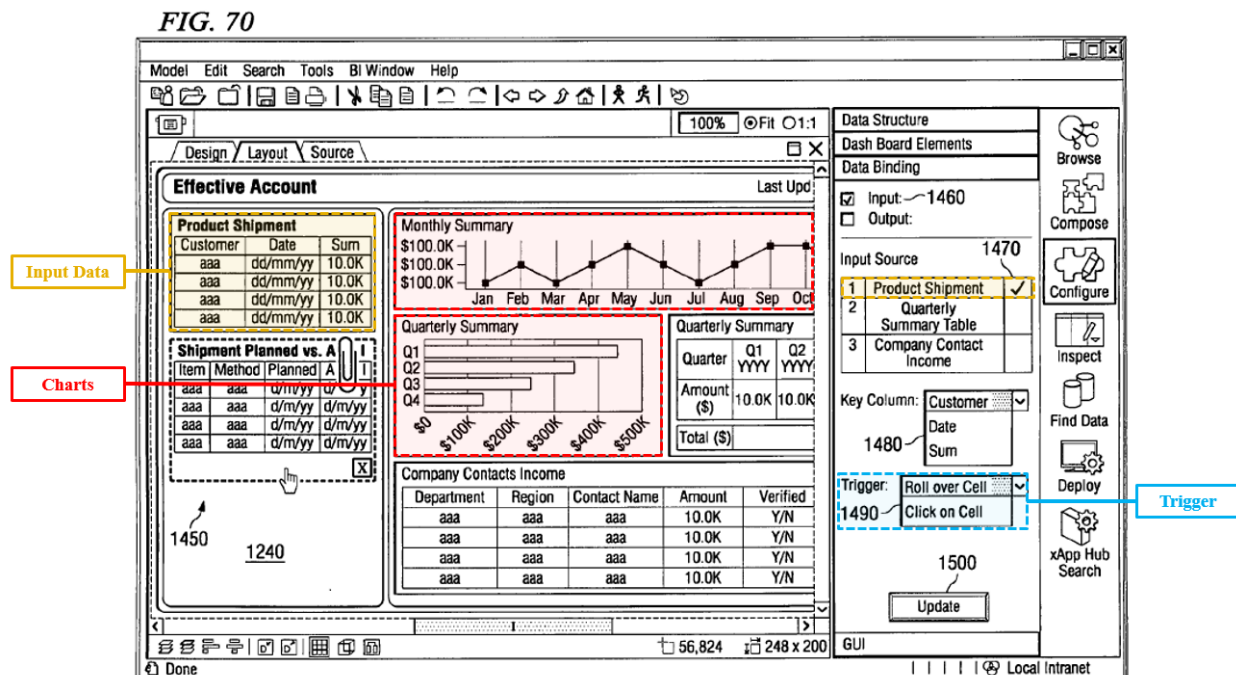
FIG. 57



EX1005, FIG. 57 (annotated).

191. 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.

192. 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).

193. 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.

194. 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.

FIG. 71

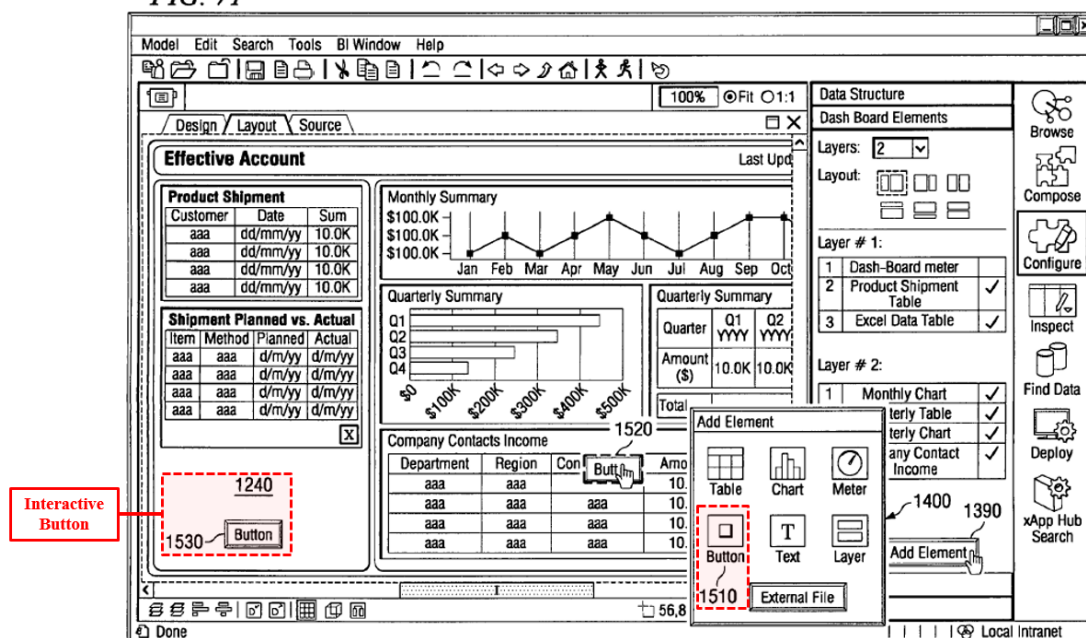
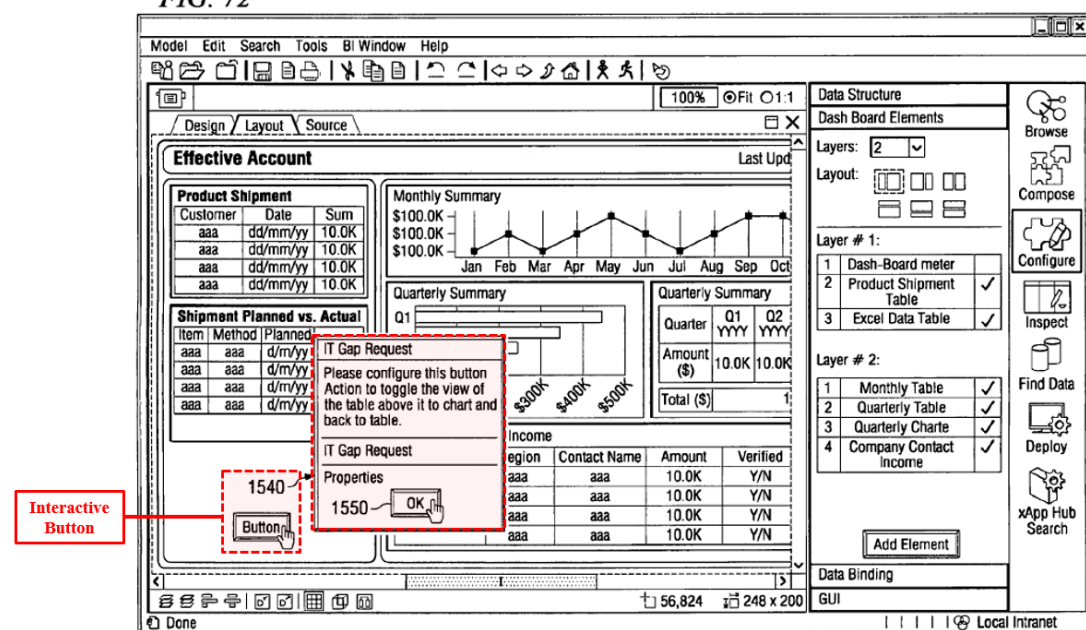


FIG. 72



EX1005, FIGs. 71, 72 (annotated).

195. It is my opinion that the types of interactive features taught by Rostoker are consistent with those disclosed in the '595 patent. As I discuss above, the '595 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, 9:25-28. And the ’595 patent discloses that these “interactive features” can include, for example, “a zoom feature, a data mouse-over popup window, a data hide/share feature, an audio playback feature, a drill down feature, a data gliding feature and the like.” EX1001, 9:28-31. As I discuss above, Rostoker’s interactive options provide similar “data hide/share” and “drill down features,” as well as additional “interactive features” that would fall within the scope of the claim. EX1001, 9:25-31.

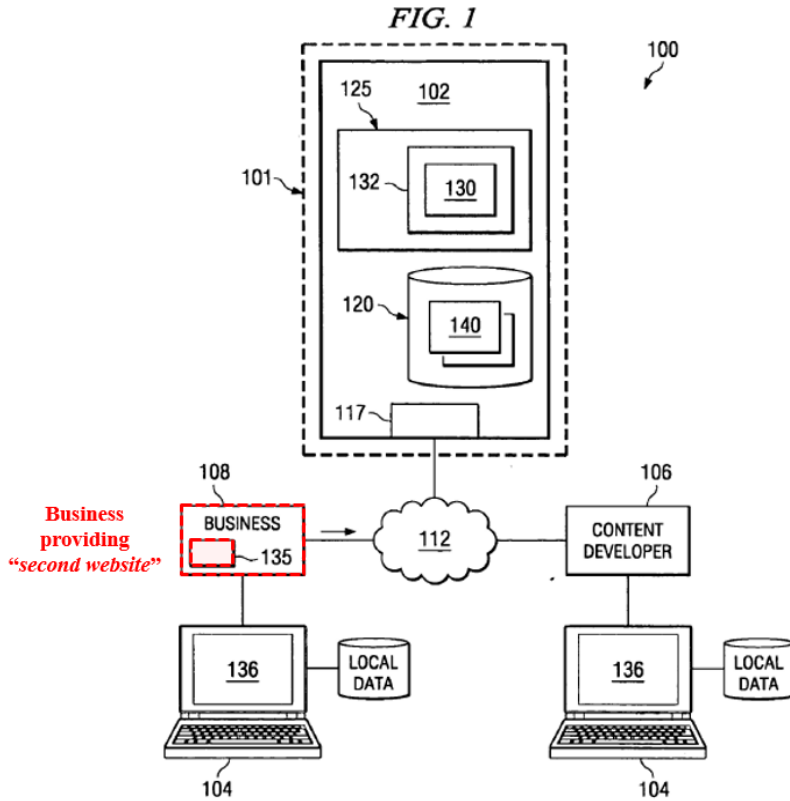
196. Therefore, it is my opinion that Rostoker teaches “*generat[ing] an interactive chart including the chart data and the chart template.*”

8. [1G] “a network interface that, in response to generation of the interactive chart, transmits the interactive chart to a second website for display without the execution of the instructions stored in the memory to generate the user interface of the first website”

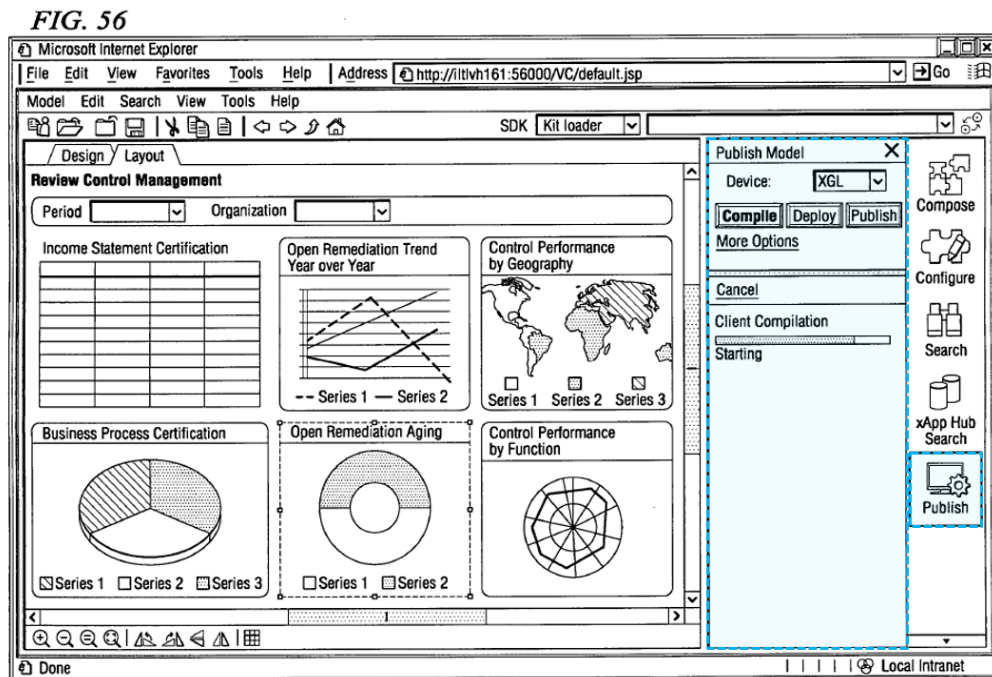
197. Element [1G] recites a processor that “*a network interface that, in response to generation of the interactive chart, transmits the interactive chart to a second website for display without the execution of the instructions stored in the memory to generate the user interface of the first website.*” In my opinion, Rostoker teaches this limitation. For ease of explanation, I address different portions of this limitation in the following subsections.

- a) **Rostoker teaches “in response to generation of the interactive chart, transmit[ting] the interactive chart to a second website for display.”**

198. Rostoker first teaches “*in response to generation of the interactive chart, transmit[ting] the interactive chart to a second website for display.*” As I discuss for element [1F], Rostoker teaches “*generat[ing] an interactive chart*” using Rostoker’s “Publish” function, for example “generat[ing] Java code [] for execution by a target device comprising a Java runtime” or “Flash code [] for execution by a target device comprising a Flash runtime.” See Section IX.B.7; EX1005, FIGs. 56, 57. Rostoker further discloses deploying and storing the application within a business environment. For example, Figure 1 of Rostoker below 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).

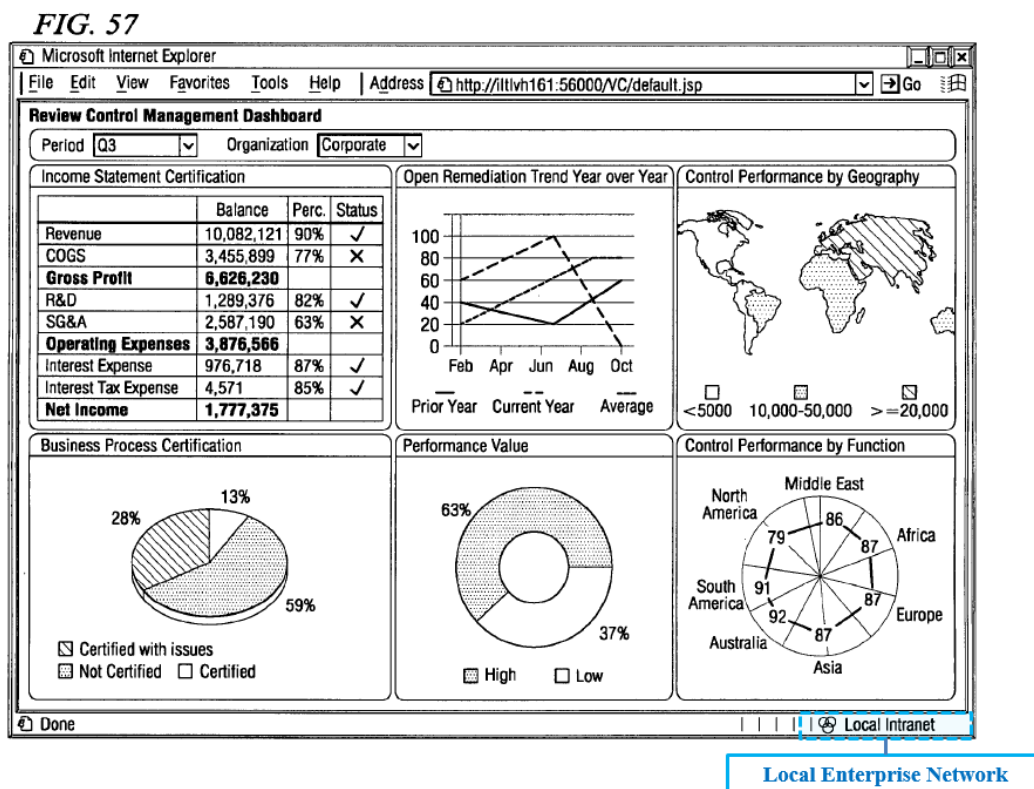


EX1005, FIG. 56 (annotated).

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, 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, ¶27 (“Users may then quickly deploy a ready-to-use analytics application that encapsulates best business practices.”). 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. It is also my opinion that 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.

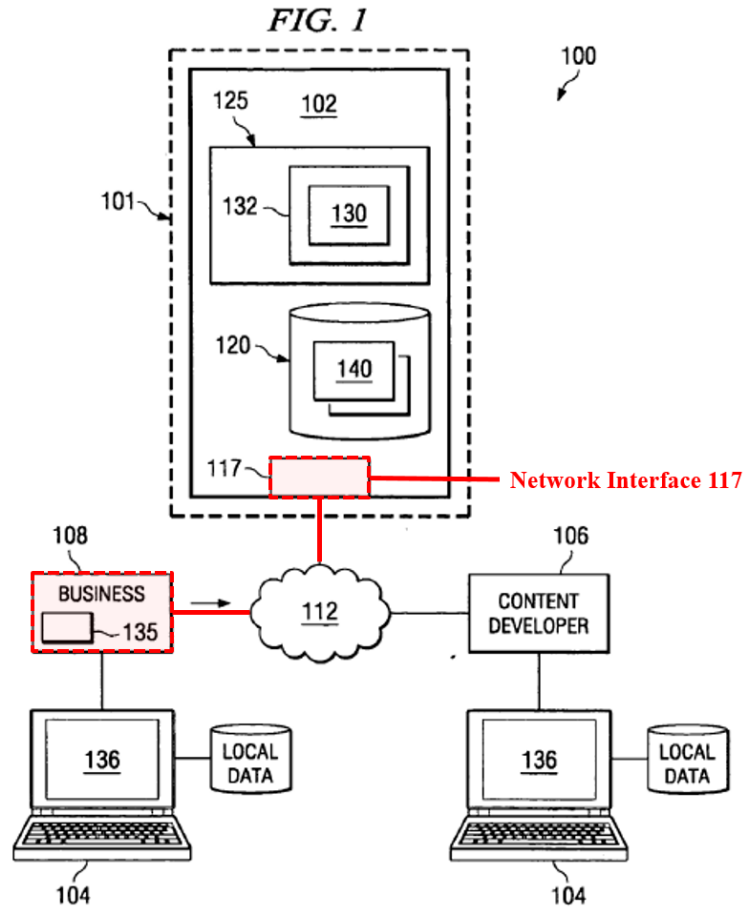
202. Moreover, providing Rostoker’s published applications “*on a second website*” would have simply been a design choice resulting in the same function of displaying the published application within a web browser. Both Intranet-based and Internet-based applications were well known and used by enterprises at the time of the ’595 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. EX10X64, 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. Additionally, because Rostoker’s “Publish” function is used to generate “*the interactive chart*” and allows the user to “Deploy” the chart, as I discuss above and for element [1F], a POSA would have understood that Rostoker’s chart would be transmitted “*to a second website*,” i.e., deployed to an enterprise network associated with business 108, “*in response to generation of the interactive chart*.” That is, after a user chooses to publish the application using

Rostoker's "Publish" functionality, as part of that process, the included "*interactive chart*" is generated and then deployed to the business's enterprise for display on "*a second website*." And because Rostoker's process generates "Java code" or "Flash code" as part of generating the chart, a POSA would have understood that once the chart would be transmitted only after that generation is complete, i.e., "*in response to generation of the interactive chart*."

b) Rostoker discloses "a network interface that . . . transmits the interactive chart to a second website"

204. It is also my opinion that Rostoker teaches "*a network interface that . . . transmits the interactive chart to a second website*." As illustrated in Figure 1 below, Rostoker's server 102 "include[s] interface 117 for communicating with other computer systems, such as clients 104, over network 112 in a client-server or other distributed environment." EX1005, ¶36. Rostoker explains that "interface 117 may comprise software supporting one or more communications protocols associated with communications network 112 or hardware operable to communicate physical signals." EX1005, ¶36.



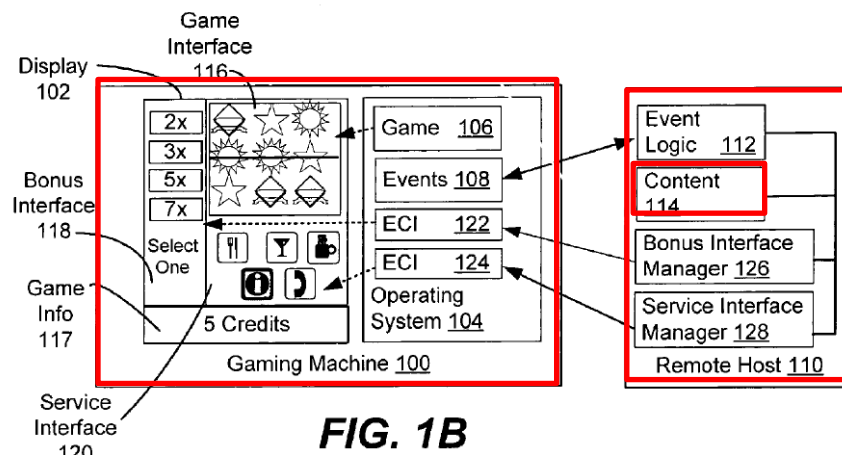
205. As I discuss in the previous section, Rostoker teaches “*transmit[ting] the interactive chart to a second website.*” Because interface 117 handles communication to and from server 102, a POSA would have understood interface 117 to perform transmission of the generated chart to business 108 via network 112, i.e., “*a network interface that, in response to generation of the interactive chart, transmits the interactive chart to a second website.*”

- c) **Rostoker discloses displaying the interactive chart on the second website “without the execution of the**

instructions stored in the memory to generate the user interface of the first website.”

206. As I explain for element [1F], Rostoker teaches publishing applications and included interactive charts as Java or Flash objects “for execution by a target device comprising” either “a Java runtime” or “a Flash runtime.” See Section IX.B.7; EX1005, ¶¶42-43, 50.

207. 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).

208. 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.

209. 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 user interface on the first website (i.e., "*without the execution of the instructions stored in the memory to generate the user interface of the first website*").

210. Accordingly, is my opinion that Rostoker discloses "*a network interface that, in response to generation of the interactive chart, transmits the interactive chart to a second website for display without the execution of the instructions stored in the memory to generate the user interface of the first website.*"

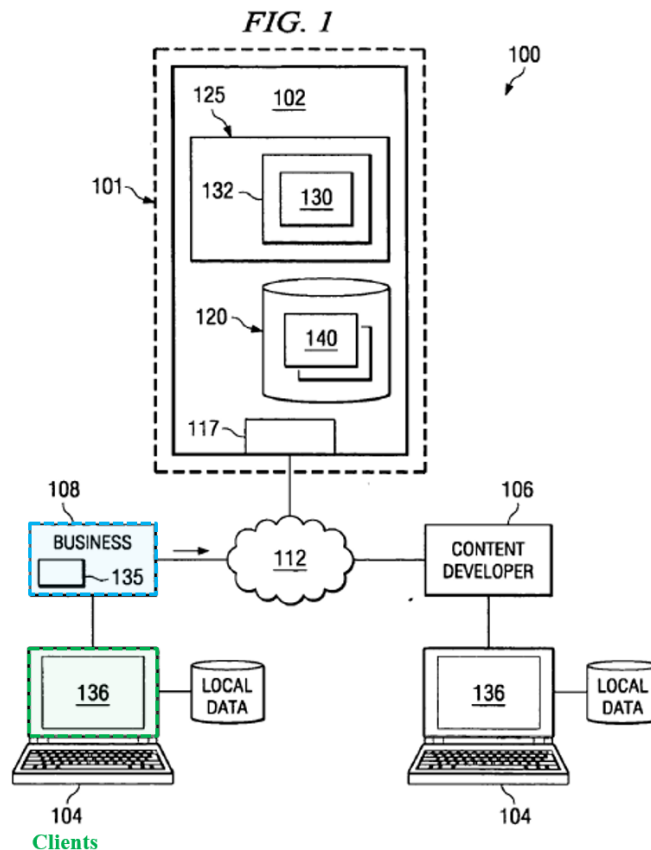
9. [1H] "wherein the second website receives an input from a second user and updates the interactive chart after the second website receives the input."

211. Element [1H] recites "*wherein the second website receives an input from a second user and updates the interactive chart after the second website receives the input.*" In my opinion, Rostoker teaches this limitation.

212. As I explain for element [1F], Rostoker teaches that its charts contain "*interactive*" features, such as changing the selection in a drop-down list, "*trigger[s]* to cause the input or output of [] data" to be associated with

components, and clicking buttons that “cause a table included with the application to switch between a table view and a chart view.” *See* Section IX.B.7; EX1005, ¶¶74, 75, FIGs. 53, 57, 70-72.

213. Figure 1 of Rostoker below illustrates a client 104 connected to business 108.



EX1005, FIG. 1 (annotated).

214. Rostoker explains that client 104, for example, “may comprise a laptop that includes an **input device**, such as a keypad, touch screen, mouse, or other device that can accept information, and an **output device that conveys information** associated with the operation of server 102 or clients 104, **including**

digital data, visual information, or GUI 136.” EX1005, ¶38. A POSA would have, therefore, understood that a client accessing the business’s published application, e.g., a business employee, would be “*a second user.*” And, because Rostoker’s published charts are interactive, as I discuss above for element [1F], Rostoker teaches allowing the “*second website [to] receive[] an input from a second user and update[] the interactive chart after the second website receives the input.*” See Section IX.B.7. As Rostoker explains, through client 104, the second user would have input devices for providing input to the business application, and an output device for viewing the output generated through interaction with the application. EX1005, ¶38.

215. Therefore, is my opinion that Rostoker discloses “*wherein the second website receives an input from a second user and updates the interactive chart after the second website receives the input.*”

B. Independent Claims 7 and 13

216. Rostoker teaches or suggests every element in claims 7 and 13. The claims are nearly identical, so I will discuss both collectively.

1. [7P] “**A method for creating interactive charts**” / [13P] “**A non-transitory computer readable storage medium having embodied thereon a program executable by a processor for creating interactive charts**”

217. Claim 7 recites “[*a*] *method for creating interactive charts.*” Claim 13 recites “[*a*] *non-transitory computer readable storage medium having embodied*

thereon a program executable by a processor for creating interactive charts.” To the extent the preambles are determined to be limiting, it is my opinion that Rostoker teaches the preambles of claims 7 and 13.

218. Particularly, Rostoker teaches the preambles of claims 7 and 13 for the same reasons it teaches the preamble of claim 1. *See* Section IX.B.1. For example, as I discuss in Section IX.B.1, 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; *see also* EX1005, claim 1 (“A computer-implemented **method** for developing an application . . .”), claim 16 (“**Software** for managing reusable business process components comprising **computer readable instructions embodied on media** . . .”). Rostoker further discloses a “[p]rocessor 125 [that] **executes instructions** and manipulates data to perform the operations of server 102.” EX1005, ¶33.

219. Therefore, Rostoker teaches the preambles of claims 7 and 13.

2. [7A]/[13A] “generating a user interface on a first website”

220. Claims 7 and 13 both recite “*generating a user interface on a first website.*” Rostoker teaches this limitation for the same reasons it teaches element [1C]. *See* Section IX.B.4.

3. [7B]/[13B] “receiving a selection of a chart template via the user interface”

221. Claims 7 and 13 both recite “*receiving a selection of a chart template via the user interface.*” Rostoker teaches this limitation for the same reasons it teaches element [1E]. *See* Section IX.B.6.

4. [7C]/[13C] “receiving entry of chart data via the user interface”

222. Claims 7 and 13 both recite “*receiving entry of chart data via the user interface.*” Rostoker teaches this limitation for the same reasons it teaches element [1D]. *See* Section IX.B.5.

5. [7D]/[13D] “generating an interactive chart including the chart data and the chart template”

223. Claims 7 and 13 both recite “*generating an interactive chart including the chart data and the chart template.*” Rostoker teaches this limitation for the same reasons it teaches element [1F]. *See* Section IX.B.7.

6. [7E]/[13E] “transmitting the interactive chart to a second website for display in response to generation of the interactive chart”

224. Claims 7 and 13 both recite “*transmitting the interactive chart to a second website for display in response to generation of the interactive chart.*” Rostoker teaches this limitation for the same reasons it teaches element [1G]. *See* Section IX.B.8.

7. [7F]/[13F] “wherein: the interactive chart is displayed via the second website without generating the user interface of the first website”

225. Claims 7 and 13 both recite “*wherein: the interactive chart is displayed via the second website without generating the user interface of the first website.*” Rostoker teaches this limitation for the same reasons it teaches element [1G]. *See* Section IX.B.8.

8. [7G]/[13G] “the second website receives an input from a second user”

226. Claims 7 and 13 both recite “*wherein . . . the second website receives an input from a second user.*” Rostoker teaches this limitation for the same reasons it teaches element [1H]. *See* Section IX.B.9.

9. [7H]/[13H] “the interactive chart is updated after the second website receives the input.”

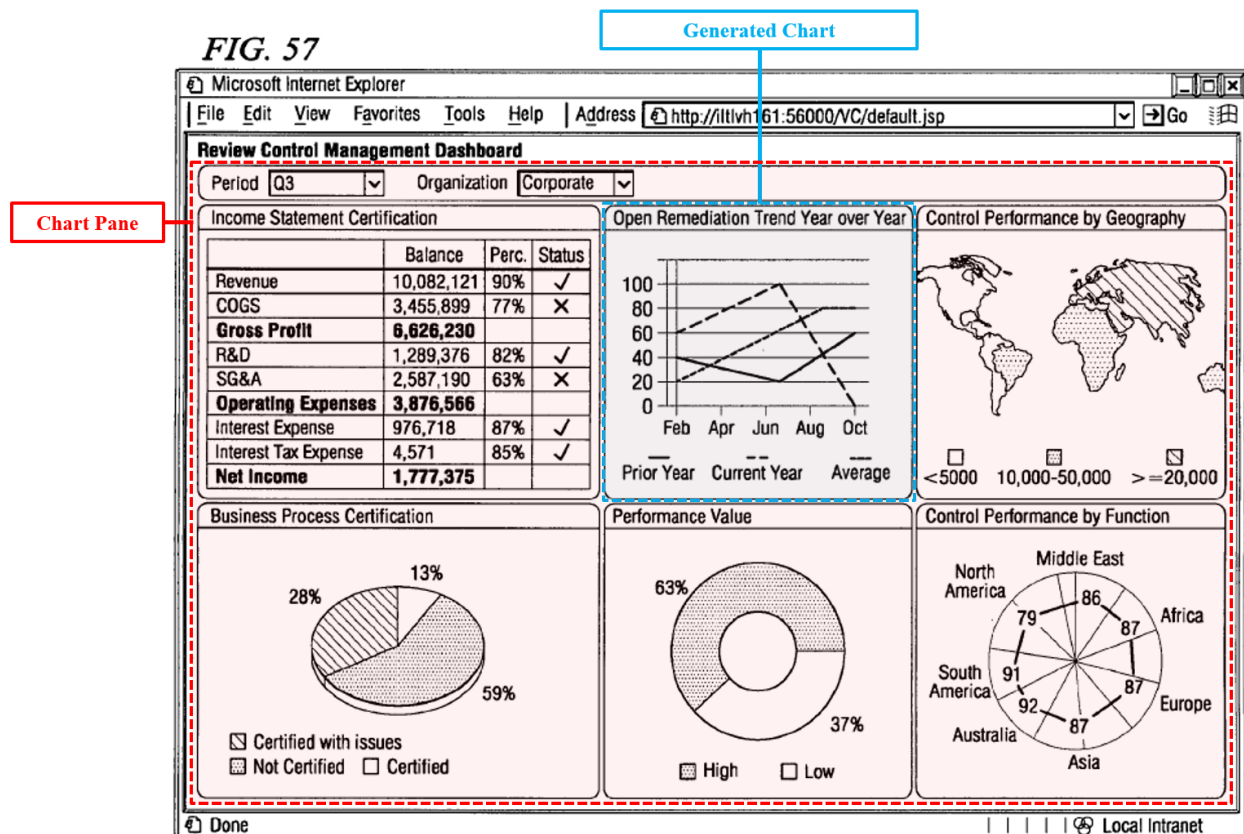
227. Claims 7 and 13 both recite “*wherein . . . the interactive chart is updated after the second website receives the input.*” Rostoker teaches this limitation for the same reasons it teaches element [1H]. *See* Section IX.B.9.

C. Claims 2, 8, and 14

228. Claims 2, 8, and 14 depend from claims 1, 7, and 13 respectively, and all recite “*wherein the user interface includes a chart pane that displays the generated interactive chart.*” In my opinion, Rostoker discloses this limitation.

229. As I explain for element [1F], Rostoker discloses generating the “*interactive chart*” when publishing the application (e.g., “generate Flash code []

for execution by a target device comprising a Flash runtime”). See Section IX.B.7; EX1005, ¶¶50, 70-71, FIGs. 56-57. As shown in Figure 57 of Rostoker, reproduced below, the “*generated interactive chart*” is displayed within “*a chart pane*” that also includes other generated charts and components.



EX1005, FIG. 57 (annotated).

230. While Rostoker suggests deploying and displaying the published application on “*a second website*,” as discussed for element [1G], Rostoker also teaches that the published application can be viewed within Rostoker’s “*user interface*.” For example, Rostoker discloses, “[i]n FIG. 56, the Publish button 1210 is selected, causing each of the xApps included in the new application to be

executed or otherwise prototyped,” and “FIG. 57 shows the new or modified application after execution.” EX1005, ¶71. Thus, Rostoker suggests that its “Publish” functionality allows applications to be viewed within Rostoker’s “*user interface*” (e.g., “executed or otherwise prototyped”). In fact, Figures 56 and 57 of Rostoker show the same URL in the “Address” bar, which suggests that the published application can be displayed in Rostoker’s “*user interface*.” Such an option would be logical, allowing developers to conveniently view the generated application, if desired, within the same interface in which the application was developed—rather than having to separately access the website to which the application was deployed.

231. For these reasons, it is my opinion that Rostoker teaches the additional limitations of claims 2, 8, and 14.

D. Claims 4, 10, and 16

232. Claims 4, 10, and 16 depend from claims 1, 7, and 13 respectively, and all recite “*wherein the interactive chart retains one or more interactive features when displayed via the second website.*” In my opinion, Rostoker discloses this limitation.

233. As I explain for element [1F], Rostoker discloses publishing charts (e.g., xApps) as Java or Flash objects “for execution by a target device comprising” either “a Java runtime” or “a Flash runtime.” *See* Section IX.B.7;

EX1005, ¶¶42-43, 50. Rostoker also discloses the published charts are “*interactive chart[s]*” with multiple “*interactive features*,” as I discuss above for element [1F]. See Section IX.B.7; EX1005, ¶¶71, 74, FIGs. 53, 57, 70-72.

234. As I also discuss for element [1G], Rostoker teaches that its published “*interactive chart[s]*” can be displayed independent of the user interface used to create the charts. See Section IX.B.8(c). That is the purpose of creating independent Java or Flash objects “for execution by a target device” using either “a Java runtime” or “a Flash runtime.” EX1005, ¶50. Therefore, a POSA would have thus understood that Rostoker’s “*interactive chart retains one or more interactive features when displayed via the second website*” because it is published/generated as a Java or Flash object that can be executed by an independent runtime— independent of the user interface used to create the chart.

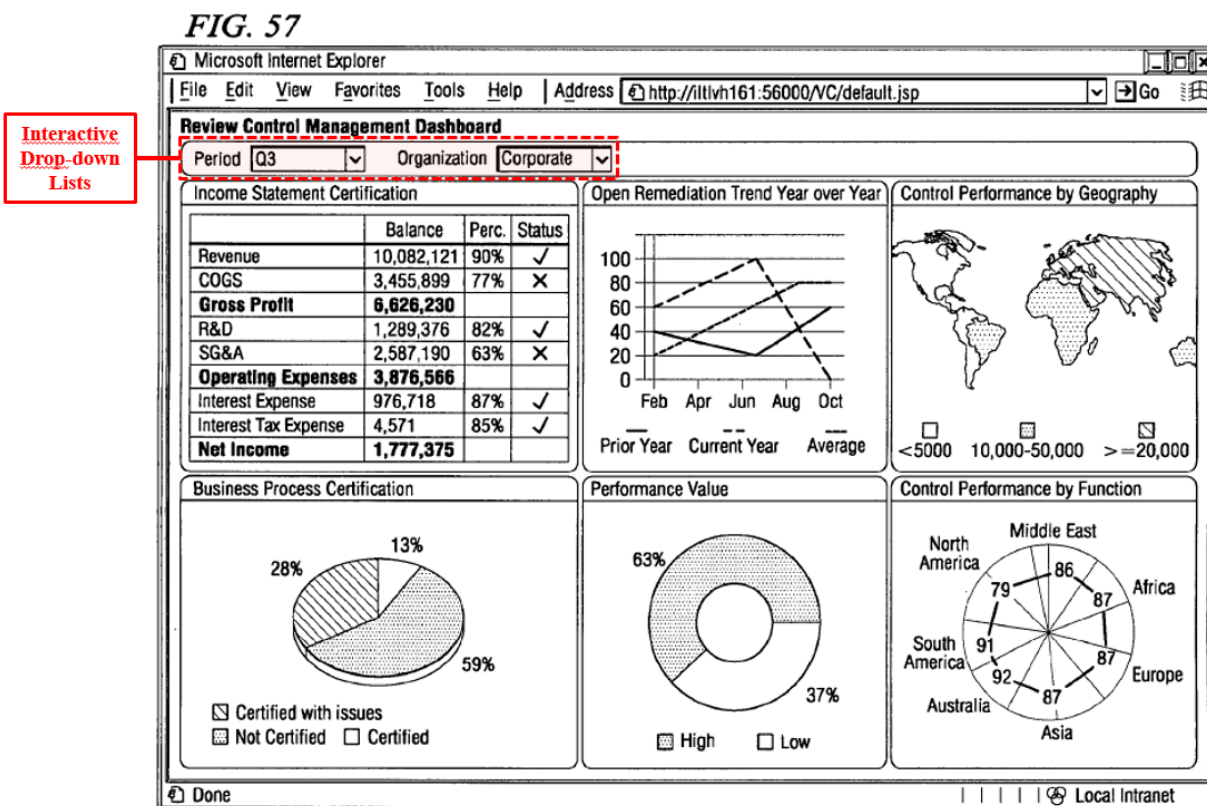
235. Therefore, it is my opinion that Rostoker teaches the additional limitations of claims 4, 10, and 16.

E. Claims 5, 11, and 17

236. Claims 5, 11, and 17 depend from claims 4, 10, and 16 respectively, and all recite “*wherein the one or more interactive features include at least one of a zoom feature, a data mouse-over pop-up window, a data hide feature, a data share feature, an audio feature, a video feature, an animation feature, and a drill-*

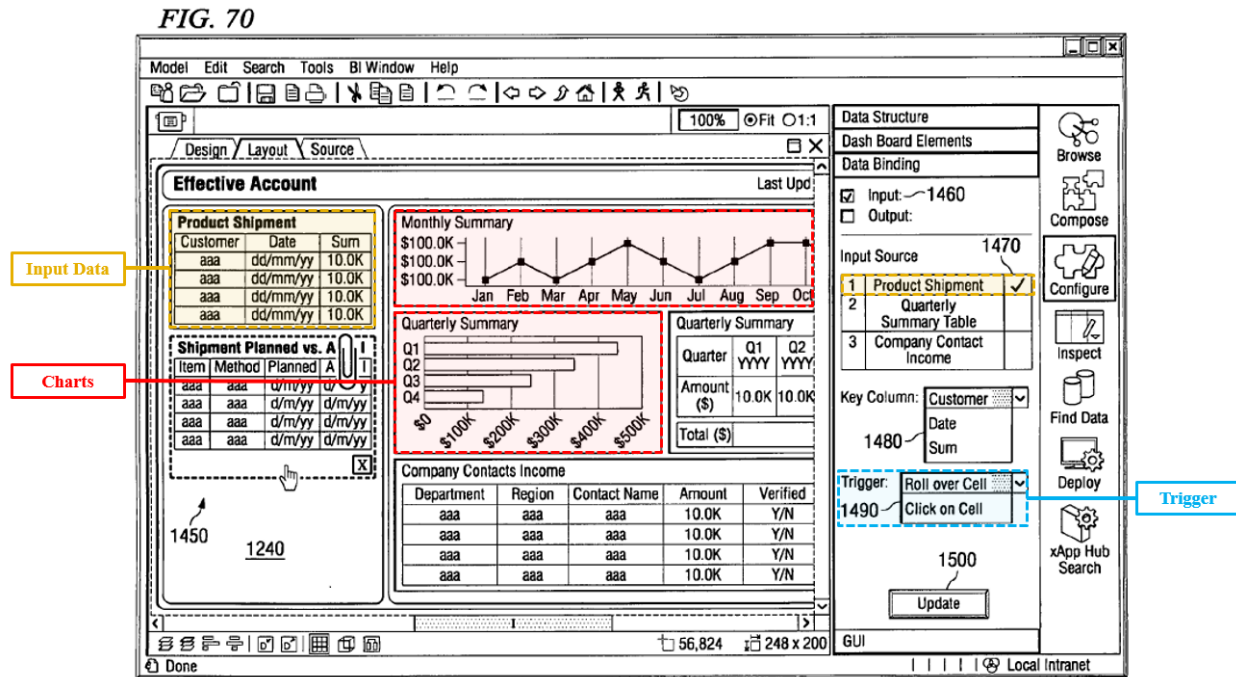
down feature, or[/and] a data gliding feature.” In my opinion, Rostoker discloses this limitation.

237. As I explain for element [1F], Rostoker discloses configuring multiple “interactive features” using the “Configure Chart View” toolbar of the user interface, including adding “Data Animation[s],” i.e., *an animation feature.*” See Section IX.B.7; EX1005, FIG. 53. Rostoker also discloses a “data hide feature,” for example, as illustrated in Figure 57 below, the interactive chart includes drop-down lists for “Period” and “Organization” that change visible data, hiding data from unselected periods or organizations. See Section IX.B.7; EX1005, FIG. 57.



EX1005, FIG. 57 (annotated).

238. Rostoker further teaches “a drill-down feature,” providing a “Click on Cell” trigger that allows a user to drill down into summary data for individual customers, as illustrated in Figure 70 below. *See* Section IX.B.7; EX1005, FIG. 70.



EX1005, FIG. 70 (annotated).

239. Therefore, it is my opinion that Rostoker teaches the additional limitations of claims 5, 11, and 17.

F. Claims 6, 12, and 18

240. Claims 6, 12, and 18 depend from claims 1, 1, and 13 respectively and all recite “wherein the interactive chart is stored in [the] memory and is selectable by a subsequent user.” In my opinion, Rostoker discloses this limitation.

241. As I discuss for elements [1C]-[1F], Rostoker discloses a user interface in the form of the “modeling environment 132,” which is used to generate

interactive charts. *See* Sections IX.B.4-7. For example, 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 or applications created or adapted by users, such as business analysts, using the modeling environment 132.” *See* Sections IX.B.4-7; EX1005, ¶40.

242. Rostoker also discloses that created components and applications (e.g., “*interactive chart[s]*”) can be made available to “*a subsequent user*” for selection via Rostoker’s hub/storefront. 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,” such as those “created or adapted by users, such as business analysts, using the modeling environment 132.” EX1005, ¶¶6, 40. And, as I discuss for element [1A], these pre-existing applications and components are stored in Rostoker’s memory 120, i.e., “*wherein the interactive chart is stored in [the] memory and is selectable by a subsequent user.*” *See* Section IX.A.2.

243. Therefore, it is my opinion that Rostoker teaches the additional limitations of claims 6, 12, and 18.

X. Ground 2: Rostoker in view of Clark renders obvious claims 1, 2, 4-8, 10-14, and 16-18.

244. Rostoker in view of Clark teaches or suggests every element in claims 1, 2, 4-8, 10-14, and 16-18.

245. As I discuss for Ground 1, it is my opinion that Rostoker renders obvious “*a network interface that, in response to generation of the interactive chart, transmits the interactive chart to a second website for display,*” as recited in element [1G] of claim 1. However, to the extent Patent Owner may argue that Rostoker does not expressly disclose transmitting its “*interactive chart*” to “*a second website,*” it is my opinion that Clark also provides this teaching.

246. 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.

247. 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.

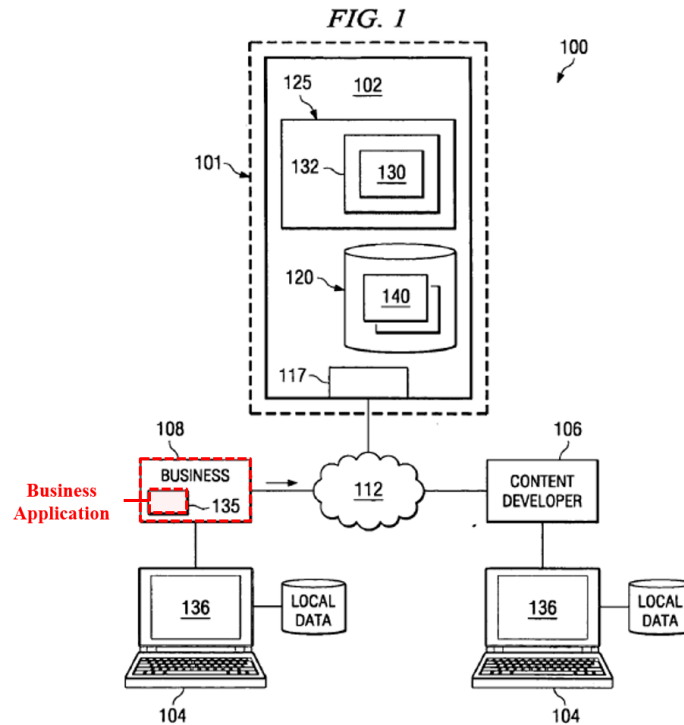
248. In this manner, Clark discloses that users can “publish” charts for display on “*a second website*,” e.g., within an “enterprise portal”: “The system 200 can **publish standalone dashboards, portals, or web pages**, as well as components **to be integrated into groupware or enterprise portals**.” 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.

249. 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 “provid[ing] the user services to publish the charts and portal pages.” EX1010, ¶106. Therefore, it is my opinion that Clark teaches publishing and displaying interactive charts “*on a second website*.”

250. 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 Clark teaches. EX1010, ¶49. Like Clark, 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 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’s system.



EX1005, FIG. 1 (annotated).

251. 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’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.

252. Moreover, techniques for providing business applications through an enterprise web server were well known at the time of the ’595 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’s system, and the combination would yield predictable results (e.g., publishing Rostoker’s interactive charts to a business website).

253. 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, 2, 4-8, 10-14, and 16-18.

XI. Grounds 3 and 4: Rostoker, alone or in view of Clark, and further in view of Jou renders obvious claims 1, 2, 4-8, 10-14, and 16-18.

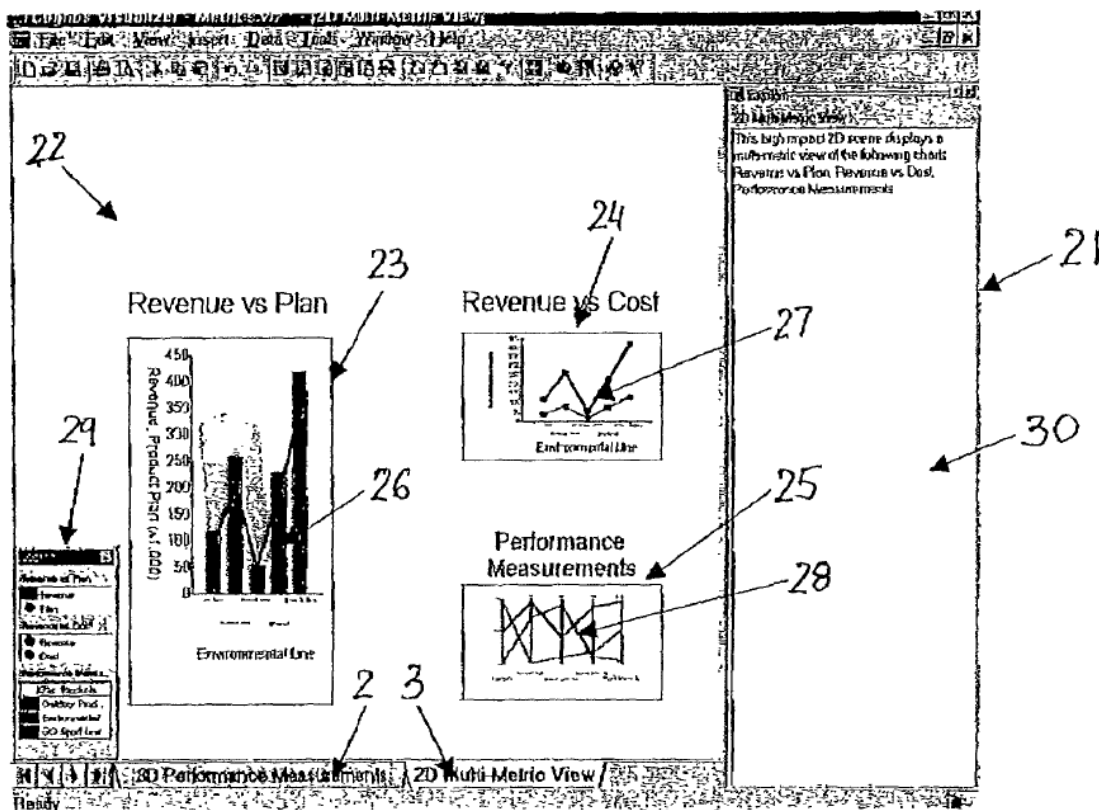
254. Rostoker, alone or in view of Clark, and further in view of Jou teaches or suggests every element in claims 1, 2, 4-8, 10-14, and 16-18.

255. As I discuss for Ground 1, Rostoker renders obvious the “*interactive*” feature limitations of elements [1F] and [1H] of claim 1, [7D], [7G], and [7H] of claim 7, [13D], [13G], and [13H] of claim 13, and claims 5, 11, and 17. *See* Sections IX.B.7, IX.B.9, IX.C.5, IX.C.8, IX.C.9, IX.F. However, to the extent it is argued that Rostoker does not expressly disclose the recited interactive features of the ’595 patent, Jou discloses such additional “*interactive*” features. It is my opinion that a POSA would have found Jou’s additional “*interactive*” features useful with Rostoker’s charts.

256. 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.

257. Jou’s charts include multiple “interactive” features, including “a data hide feature,” “a drill-down feature,” and “a zoom feature,” as recited in claims 5, 11, and 17. For example, Jou discloses various “filter[s],” only showing a subset of data and hiding other data. 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.

258. 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.

259. In my opinion, a POSA would have been motivated to incorporate Jou’s additional “*interactive*” features into Rostoker’s system (and/or the combined system of Rostoker and Clark) 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.

260. In my opinion, a POSA also would have had a reasonable expectation of success incorporating Jou’s additional interactive features into Rostoker’s system (and/or the combined system of Rostoker and Clark) 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 I discuss above for elements [1F] and [1G] in Ground 1. *See* Sections IX.B.7, IX.B.8. 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/or the combined system of Rostoker and Clark), and the combination would yield predictable

results (e.g., incorporation of additional interactive chart features into Rostoker's existing system).

261. For these reasons, and the reasons I discuss in Ground 1, the combination of Rostoker and Jou, as well as the combination of Rostoker, Clark, and Jou, render obvious claims 1, 2, 4-8, 10-14, and 16-18.

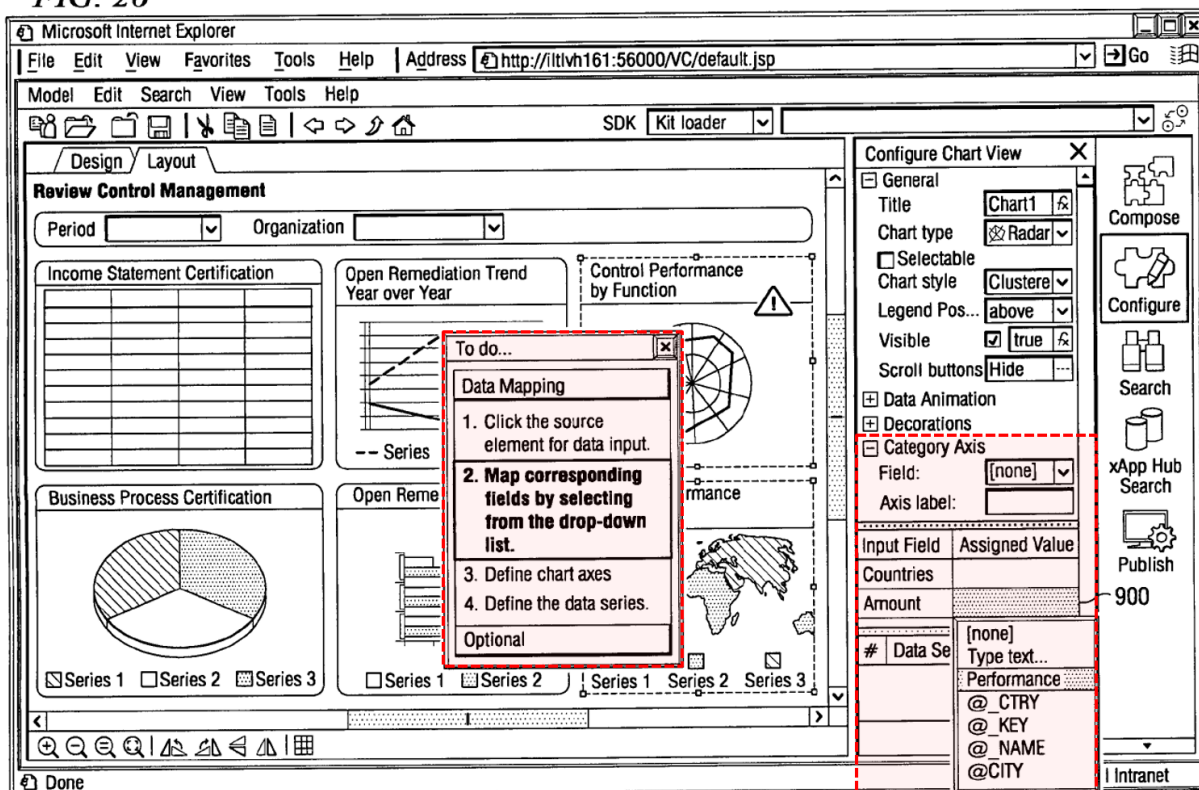
XII. Grounds 5-8: Rostoker, alone or in view of Clark and/or Jou, and further in view of Couckuyt renders obvious claims 3, 9, and 15.

A. Claim 3

262. Claim 3 depends from claim 2 and recites “*wherein the user interface further includes a data pane that allows for the first user to drag the chart data from the data pane to the chart pane, wherein the interactive chart is generated after the chart data is dragged from the data pane to the chart pane.*” In my opinion, the combinations of: (i) Rostoker and Couckuyt; (ii) Rostoker, Clark, and Couckuyt; (iii) Rostoker, Jou, and Couckuyt; and/or (iv) Rostoker, Clark, Jou, and Couckuyt each teach this limitation.

263. As I discuss in Ground 1 for element [1D] of claim 1, Rostoker discloses a “Data Mapping” function in its user interface to map data to a chart, and specific data fields may be mapped to chart axes, etc., as illustrated in Figure 26 below. EX1005, ¶68, FIG. 26.

FIG. 26



EX1005, FIG. 26 (annotated).

264. As shown in Figure 26, Rostoker illustrates a series of drop-down menus used to “Map corresponding fields,” “Define chart axes,” and “Define the data series.” EX1005, FIG. 26. A POSA would have recognized, however, that such a process can be cumbersome and less intuitive than other design options. Neither Clark nor Jou cures this deficiency. A POSA would have therefore looked for other techniques to map data to charts, such as those disclosed in Couckuyt.

265. Couckuyt discloses a “drag-and-drop user interface (DDUI) . . . associated with a data source” that allows a user to generate a chart by dragging-and-dropping data fields into “drop zones” of the chart (i.e., “a data pane that

allows for the first user to drag the chart data from the data pane to the chart pane”). EX1006, 7:10-31, FIG. 4. Couckuyt’s Figure 4 below illustrates the DDUI permitting “value data fields and the characteristic data fields . . . to be dragged and dropped into drop zones.” EX1006, 7:24-26. Couckuyt discloses that “[d]ropping a field icon into a drop zone causes the charting DDUI to populate a corresponding portion of the chart with the data from the field” (i.e., “wherein the interactive chart is generated after the chart data is dragged from the data pane to the chart pane”). EX1006, 1:32-34.

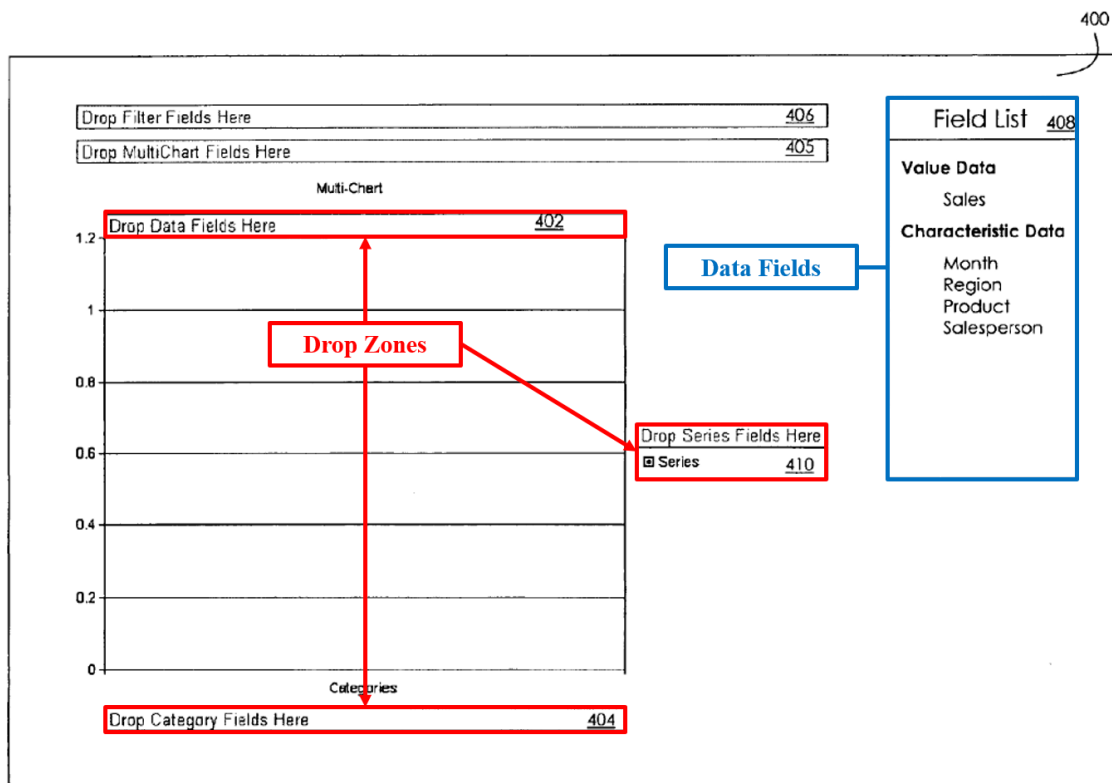
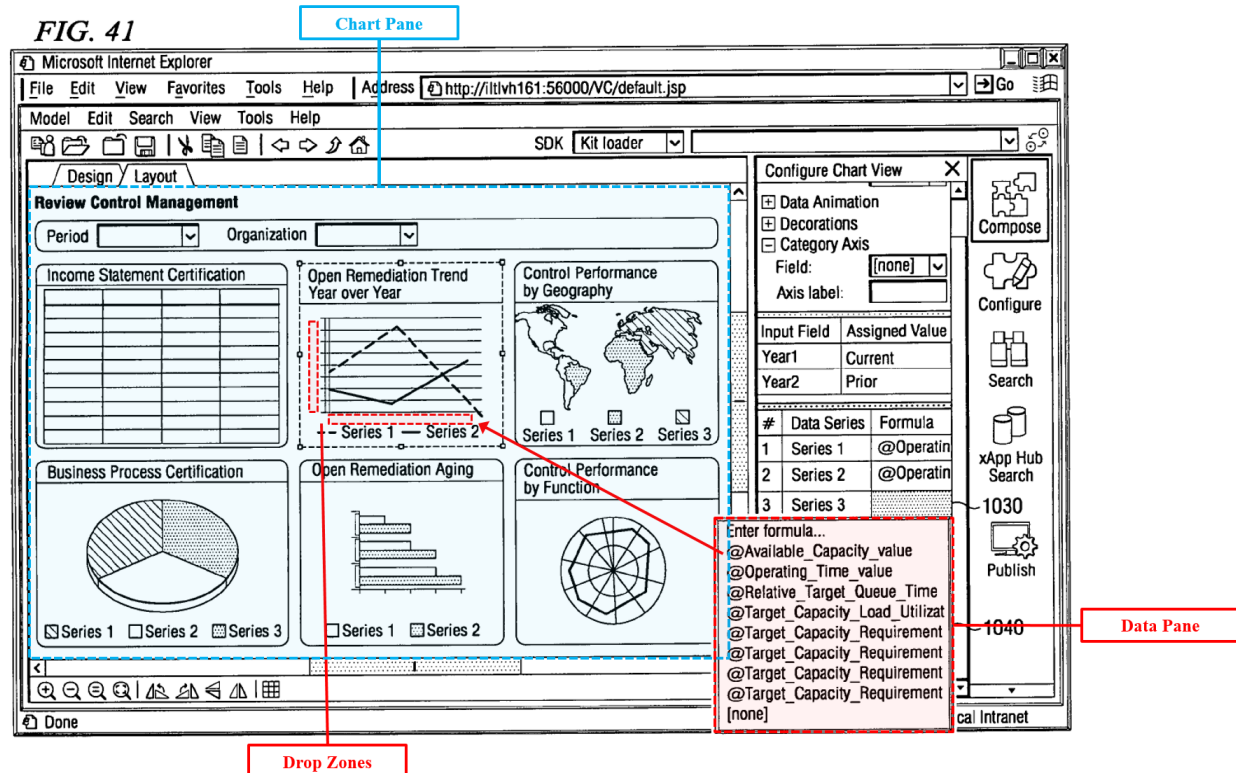


Fig. 4

EX1006, FIG. 4 (annotated).

266. It is my opinion that a POSA would have been motivated to incorporate Couckuyt's drag-and-drop techniques into Rostoker's system (and/or the combined systems of: (i) Rostoker and Clark; (ii) Rostoker and Jou; and/or (iii) Rostoker, Clark, and Jou) to enable a user to intuitively assign data to different chart aspects (e.g., different axes). For example, rather than having to select a chart and navigate through the various drop-down menus of Rostoker's "Configure Chart View," *see, e.g.*, EX1005, FIG. 41, Couckuyt's techniques would allow a user to simply select a data field and drag it to an area of the desired chart. As Couckuyt explains, a "primary benefit of the charting DDUI is that the user can quickly change the data that is being presented in the chart." EX1006, 1:18-19. Thus, using Couckuyt's techniques, a user can see the effects of the data assignment upon dragging and dropping the data, and can "quickly change the data" as desired. Moreover, Couckuyt's drag-and-drop techniques would allow a user to map data fields to any chart displayed in Rostoker's user interface, without having to individually select each chart and navigate through the "Configure Chart View" options for each chart.

267. Example functionality that would be present in the combination(s) is illustrated in Figure 41 below, which would enable the user to drag data fields directly onto a desired chart.



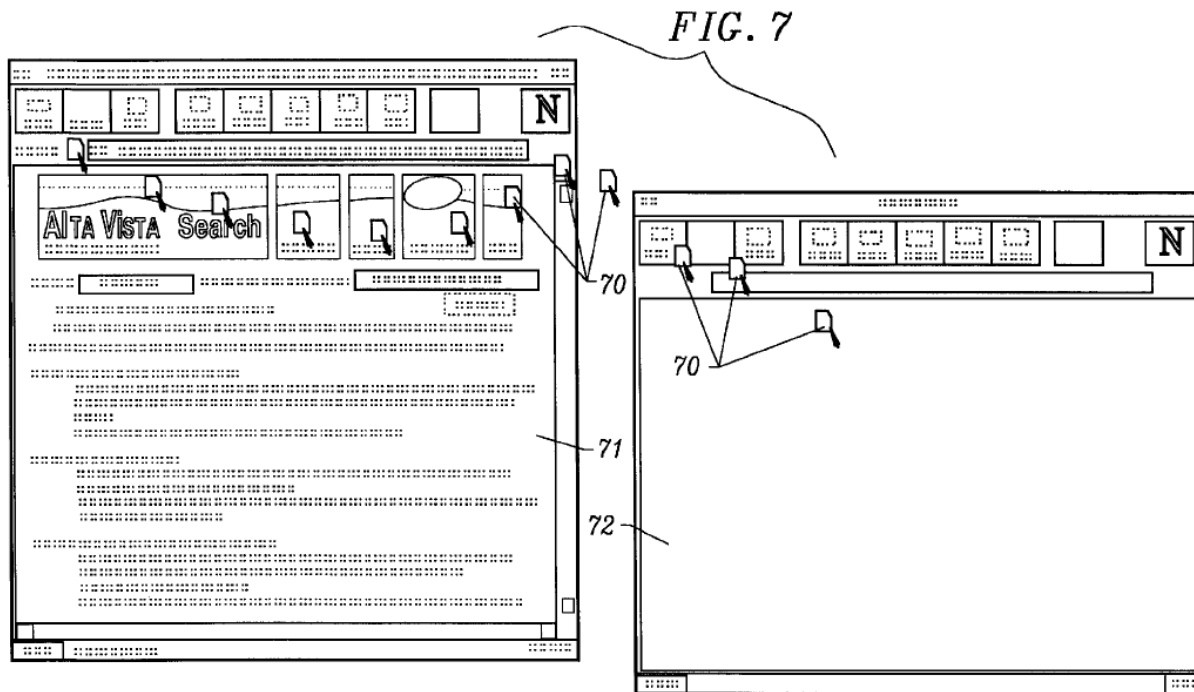
EX1005, FIG 41 (annotated).

268. For example, as illustrated in annotated Figure 41 above, Rostoker’s modeling environment includes a “data pane” and a “chart pane.” EX1005, ¶70, FIG. 41. Incorporation of Couckuyt’s drag-and-drop techniques, as I discuss above, would allow a user to drag-and-drop data fields from the “data pane” onto charts within the “chart pane,” which would update the charts with the new data, simplifying Rostoker’s data mapping process. See EX1005, FIGs. 41, 52 (showing chart updated after selection of new data); EX1006, 1:32-34 (“Dropping a field icon into a drop zone causes the charting DDUI to populate a corresponding portion of the chart with the data from the field.”). Then, as discussed for element

[1F] of claim 1, the “*interactive chart is generated*” when the chart is published using Rostoker’s “Publish” function, which would occur “*after the chart data is dragged from the data pane to the chart pane.*” The combinations that include Couckuyt, therefore, teach a user interface that “*includes a data pane that allows for the first user to drag the chart data from the data pane to the chart pane, wherein the interactive chart is generated after the chart data is dragged from the data pane to the chart pane,*” as recited in claim 3.

269. It is also my opinion that a POSA would have had a reasonable expectation of success incorporating Couckuyt’s data drag-and-drop techniques because the combination(s) would amount to nothing more than incorporating Couckuyt’s well-known drag-and-drop techniques into Rostoker’s system (and/or the combined systems of: (i) Rostoker and Clark; (ii) Rostoker and Jou; and/or (iii) Rostoker, Clark, and Jou). Rostoker already discloses “drag-and-drop” functionality in its modeling environment: “[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.**” EX1005, ¶59. Indeed, Rostoker discloses that some of the features that may be dragged are data fields: “In FIG. 49, **a selected data field 1140 (‘Available_Capacity_value’) is selected and dragged 1150** to the variable 1110.” EX1005, ¶70.

270. Moreover, drag-and-drop techniques were widely available and well known before the '595 patent. The Macintosh operating system, for example, included the ability to drag and drop document icons. EX1036, 0012-14. Additionally, Parsons, published in 2000 discloses dragging an icon of a source view onto a target view, such that when the source view is selected, its contents are shown in the target view. EX1047, 6:38-44. As depicted below in Figure 7, an icon 70 is dragged from source view 71 onto target view 72. EX1047, 7:16-20.



EX1047, FIG. 7.

271. Similarly, Kung, published in 2006, discloses drag and drop to transfer files between computing devices. EX1046, 7:15-20. Thus, in my opinion, a POSA would not have encountered any unreasonable technical hurdles

incorporating Couckuyt's techniques of dragging and dropping data fields directly onto a chart in Rostoker's system.

272. A POSA would have been further motivated to combine Rostoker (alone or in combination with Clark and/or Jou) with Couckuyt because both are in the same general field of generating interactive charts and address the same problem of efficiently generating multiple charts through a user interface. EX1005, ¶4; EX1006, 1:56-2:2. As I discuss above, modifying Rostoker's system (and/or the combined systems of: (i) Rostoker and Clark; (ii) Rostoker and Jou; and/or (iii) Rostoker, Clark, and Jou) to use Couckuyt's well-known data drag-and-drop techniques would merely amount to combining prior-art elements (Rostoker's modeling environment for generating interactive charts and Couckuyt's drag-and-drop techniques) according to known methods, yielding predictable results (e.g., allowing drag-and-drop selection of data).

273. Therefore, it is my opinion that the combinations of: (i) Rostoker and Couckuyt; (ii) Rostoker, Clark, and Couckuyt; (iii) Rostoker, Jou, and Couckuyt; and/or (iv) Rostoker, Clark, Jou, and Couckuyt each renders obvious claim 3.

B. Claim 9

274. Claim 9 depends from claim 8 and recites “*receiving an indication that the chart data has been dragged from a data pane in the user interface and dropped into the chart pane; and generating the interactive chart after the chart*

data is dropped into the chart pane.” It is my opinion that the combinations of: (i) Rostoker and Couckuyt; (ii) Rostoker, Clark, and Couckuyt; (iii) Rostoker, Jou, and Couckuyt; and/or (iv) Rostoker, Clark, Jou, and Couckuyt each renders obvious claim 9 for the reasons I discuss for claim 3. *See* Section XII.A. As a POSA would have understood, the system would receive “*an indication*” that that the drag and drop occurred through conventional mechanisms (e.g. clicking and releasing of a mouse). *See* Section VI.D (describing conventional drag-and-drop).

C. Claim 15

275. Claim 15 depends from claim 14 and recites “*wherein the program is further executable to: allow the first user to drag the chart data from a data pane of the user interface and drop the data into the chart pane; and generate the interactive chart after the chart data is dropped into the chart pane.*” It is my opinion that the combinations of: (i) Rostoker and Couckuyt; (ii) Rostoker, Clark, and Couckuyt; (iii) Rostoker, Jou, and Couckuyt; and/or (iv) Rostoker, Clark, Jou, and Couckuyt each renders obvious claim 15 for the reasons I discuss for claim 3. *See* Section XII.A.

XIII. Ground 9: The QlikView Manuals render obvious claims 1, 2, 4-8, 10-14, and 16-18.

276. The QlikView Manuals teach or suggest every element in claims 1, 2, 4-8, 10-14, and 16-18.

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

277. 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.

278. 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.

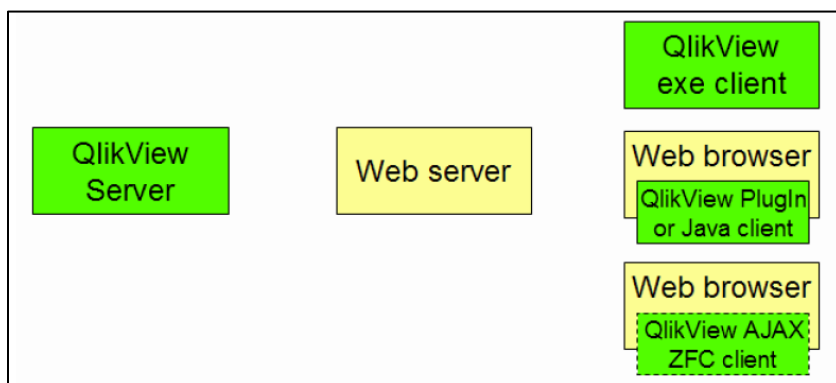
B. Independent Claim 1

279. The QlikView Manuals teach or suggest every element in claim 1.

1. [1P] “A system for creating interactive charts”

280. Claim 1 recites “[a] system for creating interactive charts.” To the extent the preamble is determined to be limiting, the QlikView Manuals disclose the preamble.

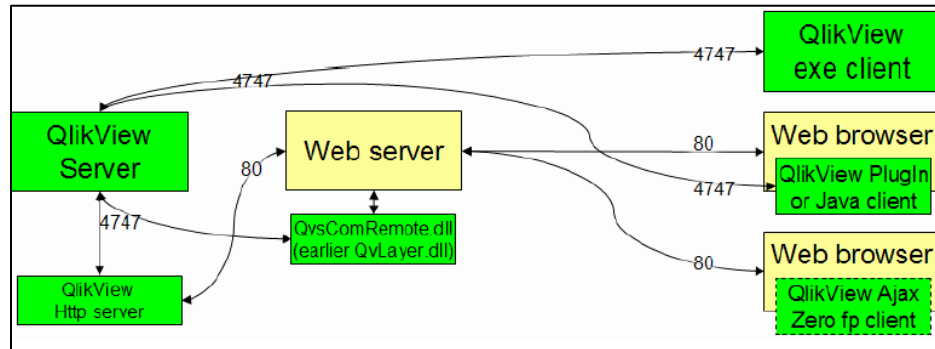
281. The QlikView Manuals describe a “system” having a QlikView Server, a Web server, and clients. EX1009, 0095.



EX1009, 0095; *see also* EX1009, 0098.

282. The figure below illustrates network connections within the QlikView system. For example, Windows Clients running QlikView directly communicate

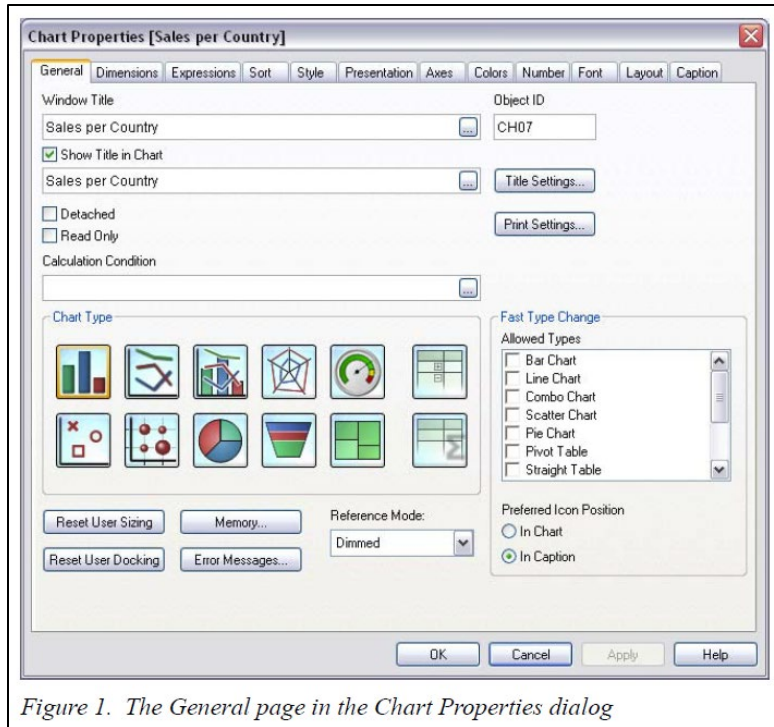
with the QlikView Server, whereas Java and AJAX clients use a web server to facilitate communication with the QlikView Server. EX1009, 0098.



EX1009, 98.

283. The QlikView Manuals describe a system for “creat[ing] dynamic graphical charts” EX1008, 0013. This allows the user to create end user interfaces and presentations based on data, perform statistical analysis, and merge information from several sources. EX1008, 0013.

284. In QlikView, “[c]harts are created by choosing New Sheet Object from the Layout menu, or by pressing the Create Chart button” EX1008, 0667. “This opens a wizard that will help you create the chart.” EX1008, 0667.



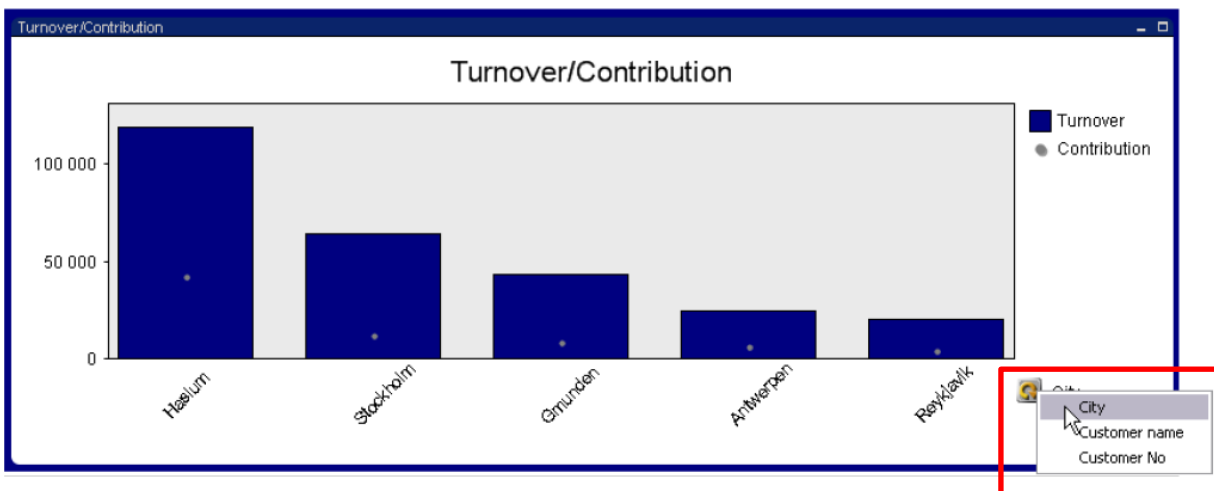
EX1008, 0882; *see also* EX1008, 0085, 0880, 0890, 1104-14.

285. For example, the user can select the Layout Menu and select Quick Chart Wizard from the dropdown to “create bar, line and pie charts” EX1008, 0084-85. QlikView also provides a Quick Chart Wizard “for the user who wants to create a simple chart in a quick and easy way.” EX1008, 1104. Using the Quick Chart Wizard the user can: (1) select a chart type; (2) define chart dimensions; (3) define a chart expression; and (4) format the chart. EX1008, 1104. The Quick Chart Wizard is also accessible from the Layout menu, as well as from the toolbar. EX1008, 1104.

286. QlikView charts are “*interactive*” because they are “dynamic.” EX1008, 0013. 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.

287. 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).

288. The charts also support pop-up labels showing data point values when the user moves the cursor over the data point. EX1008, 0919. QlikView charts

include a “Fast Type Change” feature allowing the user to select and convert the chart to a different chart type. EX1008, 0668-69. Additional interactive features include interactive search feedback, sort indicators within tables, and drop-down select within tables. EX1009, 0016-17.

289. The QlikView Manuals also disclose interactive slider objects and buttons. EX1008, 0036. For example, when a chart includes an animation, an animation bar is displayed in the chart area, including a play/pause button, and a progress bar slider allowing the user to click and move through animation frames. EX1008, 0899-0900.

290. 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.

291. The interactive features described in the QlikView Manuals are similar to those described in the '595 patent. The '595 patent describes interactive features including “a zoom feature, a data mouse-over popup window, a data hide/share feature, an audio playback feature, a drill down feature, a data gliding feature and the like.” EX1001, 9:28-31. The '595 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, 9:25-28. As discussed above, QlikView’s interactive options provide similar interactive features.

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

2. [1A] “memory that stores a chart template.

293. Claim 1 recites “*memory that stores a chart template.*” The QlikView Manuals disclose this limitation.

294. The QlikView Manuals disclose this limitation because they explain how QlikView stores chart themes. EX1008, 0795-0805, 1323. Like the templates in the ’595 patent, EX1001, 12:51-53, QlikView themes maintain a consistent “look and feel” for charts:

theme: A file containing the properties and look and feel of a document, sheet or object. Themes can be created and used from inside Qlikview.

EX1008, 1323. Indeed, QlikView calls its themes “templates.” *See* EX1008, 0799.

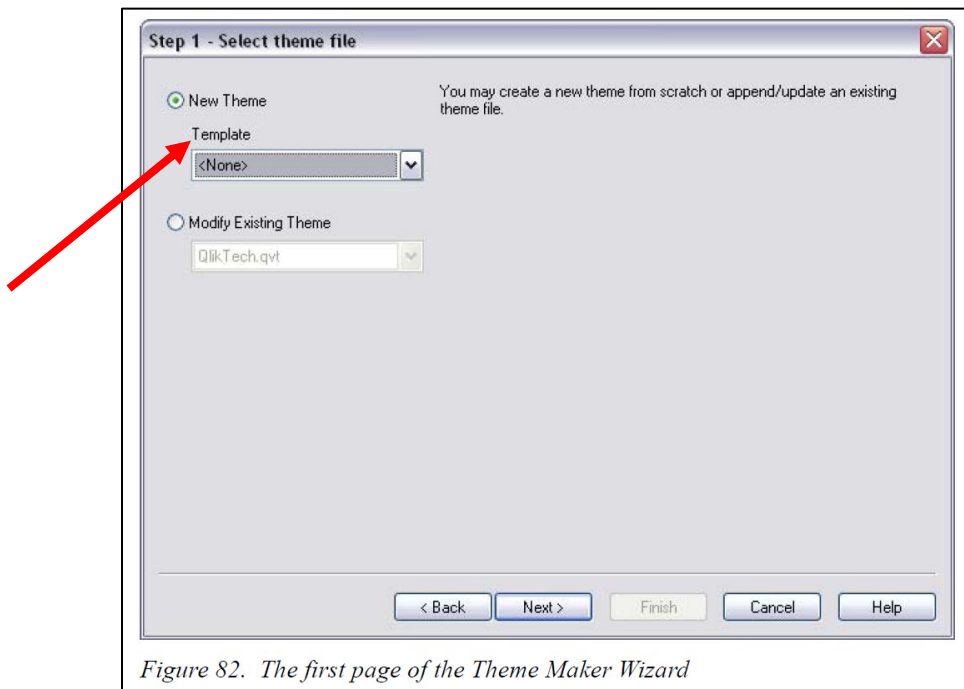
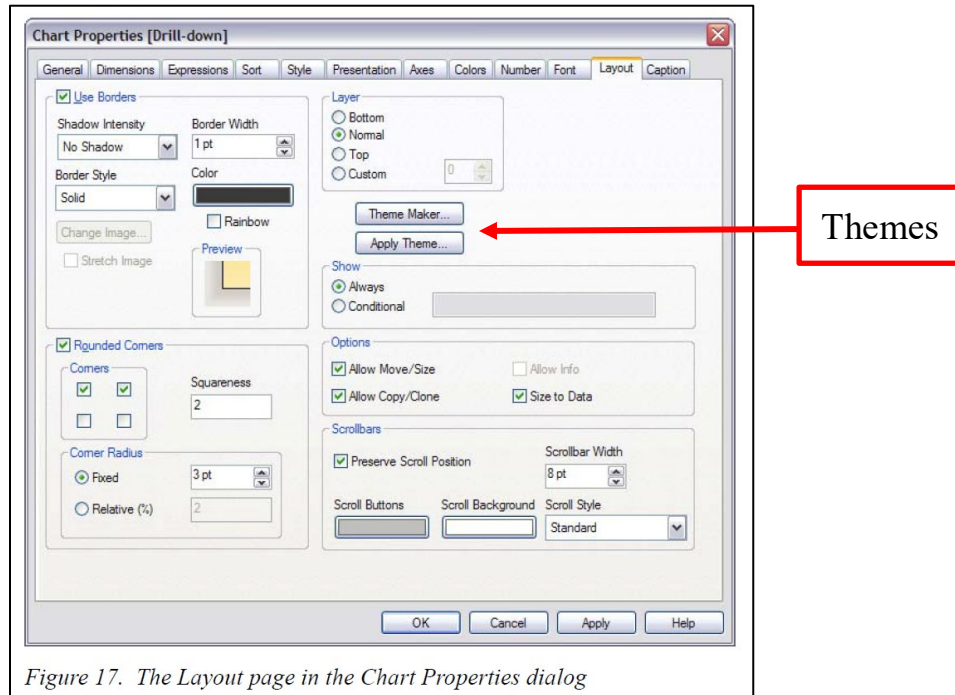


Figure 82. The first page of the Theme Maker Wizard

EX1008, 0799 (annotated).

295. QlikView's Theme Maker Wizard, shown below, creates themes/templates. EX1008, 0798-0805.



EX1008, 0939 (annotated).

296. 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.

297. 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. The user can then choose to apply the theme to the current document or all new documents. EX1008, 0804.

298. Once created, the Theme Maker Wizard saves the chart theme/template in memory. EX1008, 0795, 0804. The memory is local to the client in either “a hard disk with at least 150 MB of free disk space” or “primary memory.” EX1008, 1327; EX1009, 0025-26 (system requirements). However, the templates could be stored in memory on a remote server. *See* EX1008, 0591 (theme files reside in different locations). As stated above, when saving the theme the user can elect to apply the theme to the current document, or as a default theme for all new documents. Here, the QlikView Manuals state that “the selected theme must be accessible from disc” to be used. EX1008, 0804. Storing a theme on the disc discloses storing the theme in memory because the disc is a type of memory.

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

3. [1B] “a processor, wherein the processor executes instructions stored in the memory, causing the processor to:”

300. Claim 1 recites “*a processor, wherein the processor executes instructions stored in the memory, causing the processor to [perform certain steps].*” The QlikView Manuals disclose this limitation.

301. The QlikView Manuals disclose “a Pentium II processor or higher. (Pentium ®4 or better recommended.)” EX1008, 1326. QlikView x64 requires a “[p]rocessor supporting the x64 architecture (includes AMD Athlon 64, AMD Opteron 64, Intel Xeon EM64T and Intel Pentium with x64 extension.” EX1008, 1327. As a POSA would have understood, the Start.exe program installs instructions in the client computer’s local memory, which enables the processor to perform steps [1C]-[1F]. EX1008, 0055-59. This is the same local memory recited in [1B]. EX1008, 1326-28; EX1009, 0025-26.

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

4. [1C] “generate a user interface on a first website”

303. Claim 1 recites a processor that “*generate[s] a user interface on a first website.*” The QlikView Manuals disclose this limitation.

304. Shown below, web clients generate a user interface (UI) having a plurality of sheet objects, including charts. EX1009, 0097. The QlikView Manuals state that “[c]harts and tables are sheet objects” EX1008, 0880. Shown below,

the UIs are generated on “*a first website*” (e.g., a URL that starts with <http://localhost/QvJava/>). As a POSA would have understood, the clients generate these interfaces using a processor. EX1008, 1326-28; EX1009, 0025-26.

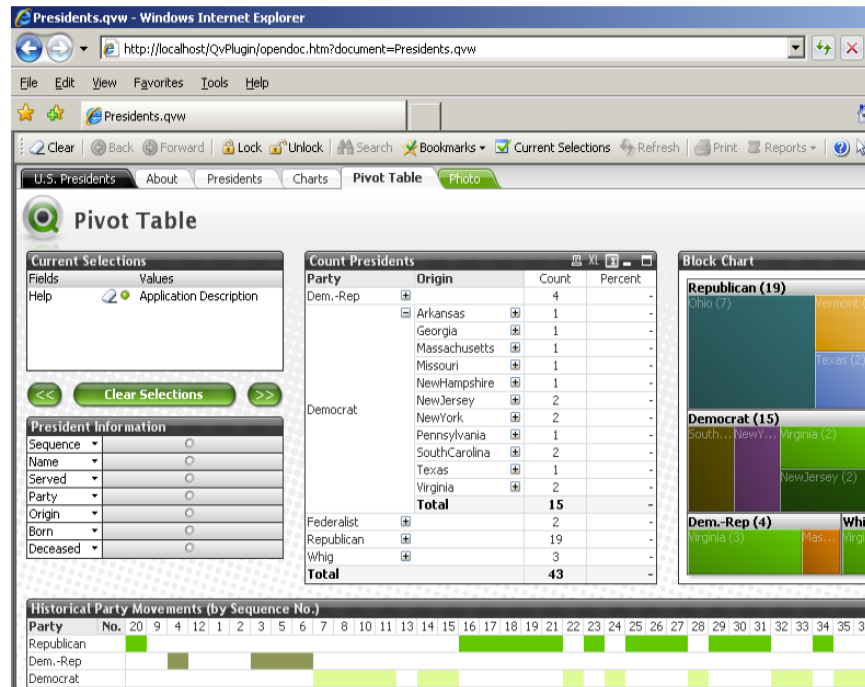
305. For example, the image below depicts “*a first website*” (e.g., <http://localhost/QvJava/opendoc.htm?document=Presidents.qvw>) with different types of sheet objects (charts) within a QlikView Java client.



QlikView Java client in Internet Explorer.

EX1009, 0129.

306. Similarly, the screenshot below also depicts “*a first website*” (e.g., <http://localhost/QvJava/opendoc.htm?document=Presidents.qvw>) with different types of sheet objects (charts) within QlikView Internet Explorer plugin client.



QlikView plug-in client in Internet Explorer.

EX1009, 0139.

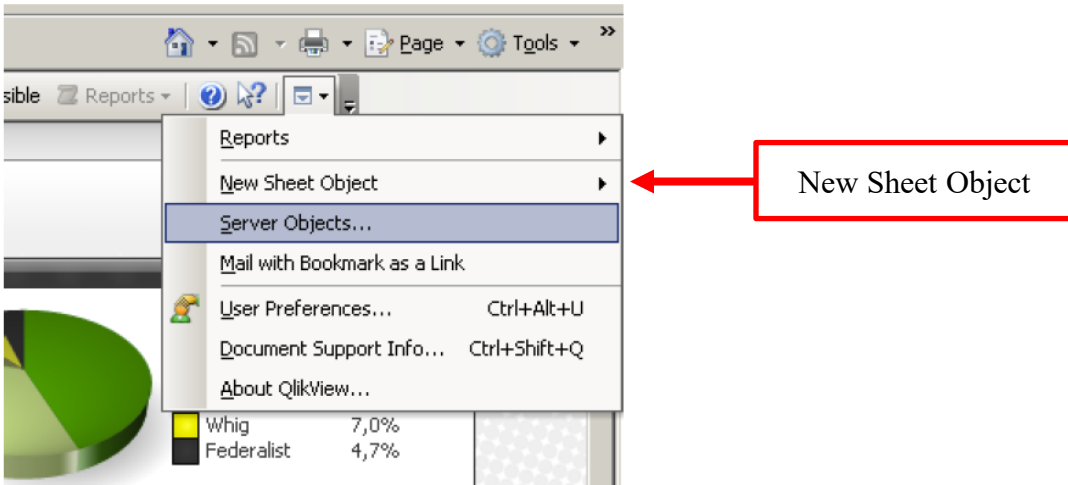
307. For at least these reasons, the QlikView Manuals disclose [1C].

5. [1D] “receive chart data via the user interface”

308. Claim 1 recites a processor that “receive[s] chart data via the user interface.” The QlikView Manuals disclose this limitation.

309. The QlikView Manuals disclose this limitation because the QlikView UI provides many ways for the processor to receive chart data. Shown below, the UI has an option to create a “New Sheet Object.” EX1009, 0140. Sheet objects include charts and chart data. EX1008, 0880. As the QlikView Manuals state, “[c]harts are created by choosing New Sheet Object” EX1008, 0667, 0880. The New Sheet Object option is available both in the Layout menu and via the

Create Chart button in the QlikView toolbar. EX1008, 0667. The user can also use the Create QuickChart option available from the list box object menu, Layout menu, or toolbar. EX1008, 0667.



EX1009, 0140; *see also* EX1008, 0022.

310. When creating a chart, the processor (e.g., the processor on a client computer) receives fields/groups and dimensions (data):

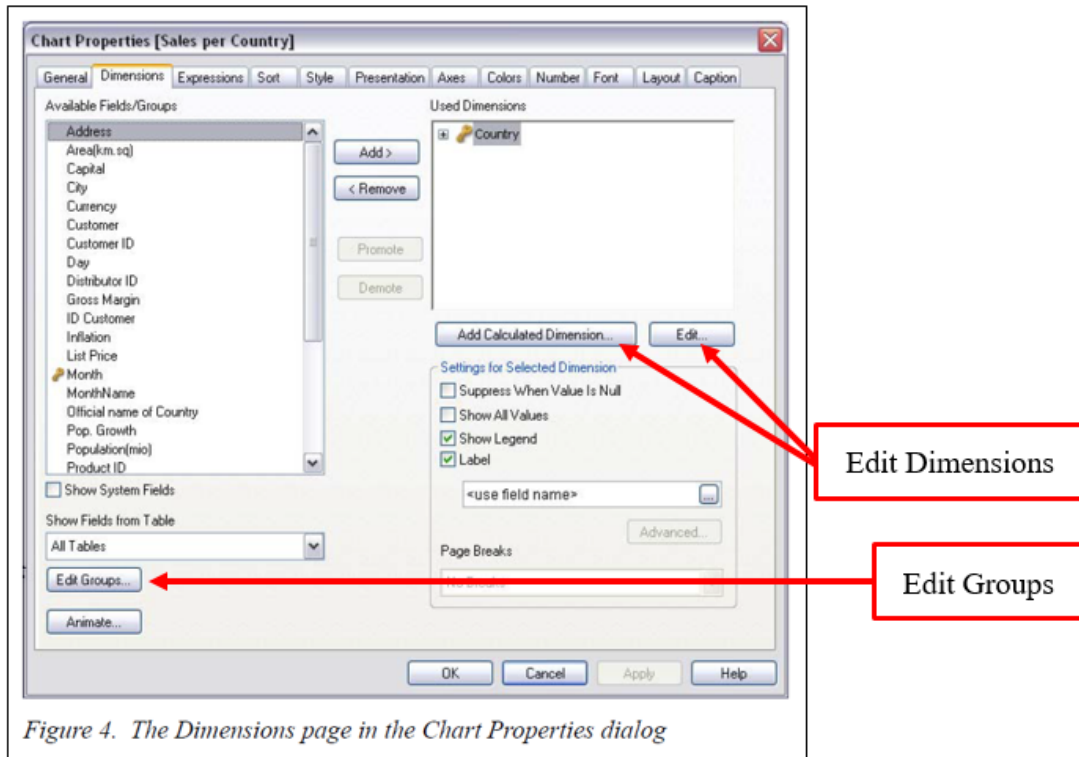


Figure 4. The Dimensions page in the Chart Properties dialog

EX1008, 0894 (annotated).

311. When a user edits groups, the processor receives new field groups (data) and stores the data within the QlikView file's data structure. EX1008, 0583-84, 0895. For example, as shown below, the user has defined a field group labeled Time, including data such as Year, Quarter, and Month. EX1008, 0582.

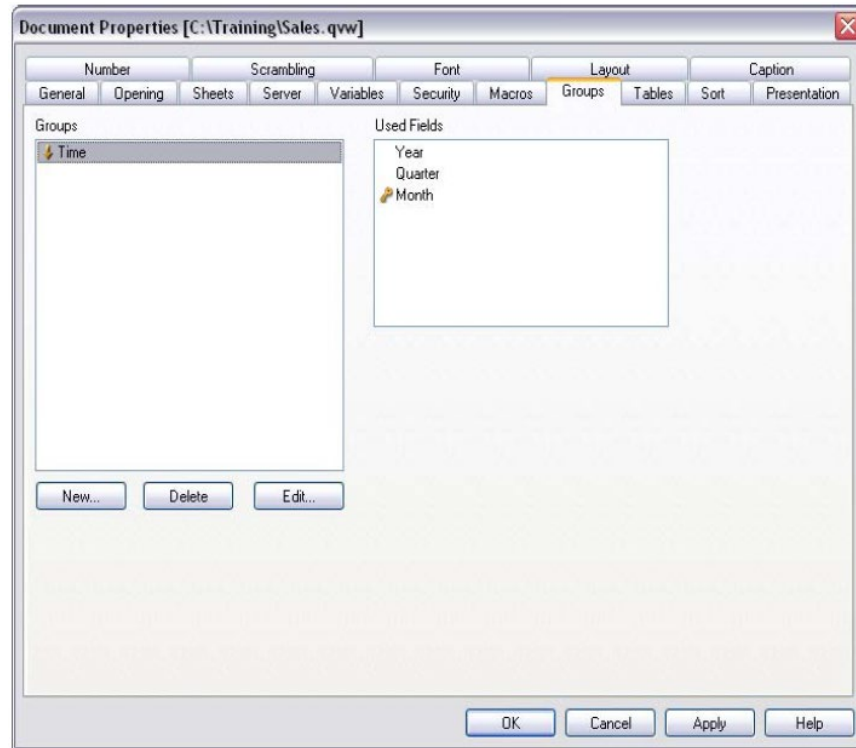


Figure 10. The Groups page in the Document Properties dialog

EX1008, 0582.

312. The processor also receives data when a user adds or edits dimensions. EX1008, 0897, 1094-97. In QlikView, charts iterate over dimensions when it calculates expression values. EX1008, 1315. Expressions are formulas (e.g., the sum, average, median, etc.) to create data points. EX1008, 1094, 1315. For example, in a bar chart an expression is used to calculate the y-axis value of a column. EX1008, 0901. Expressions are also used to define how data is displayed. For example, expressions can set the background color, text color, and text format of a data point based on the data point's value. EX1008, 0902-03. Users can add expressions (data) to the QlikView dataset here:

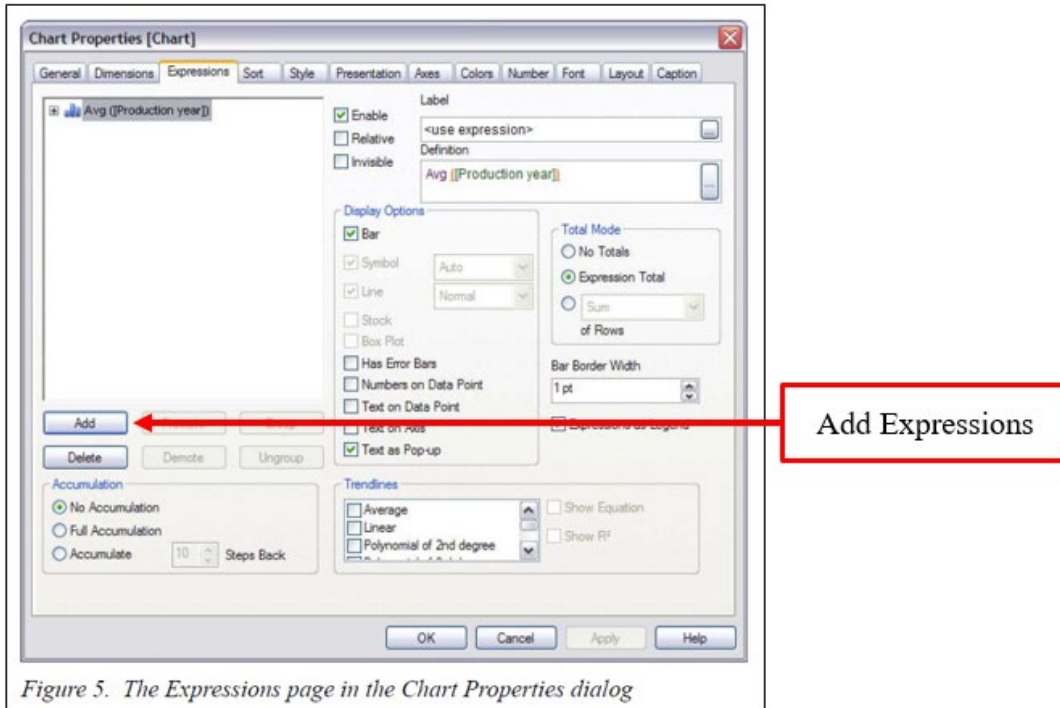


Figure 5. The Expressions page in the Chart Properties dialog

EX1008, 0901 (annotated).

313. The processor also receives data from a file or database when a script executes. EX1008, 1321. The following figure shows a script where the processor (e.g., a processor on the client computer) receives data. The fields “Id, Period, Clients, Training, Holiday, Consulting, Internal, Full” are received from an Excel file titled “ProjectTransQV7.xls” on a user’s local C: drive and transmitted to the QlikView file database during “Load.” EX1008, 0229, 0469-471.

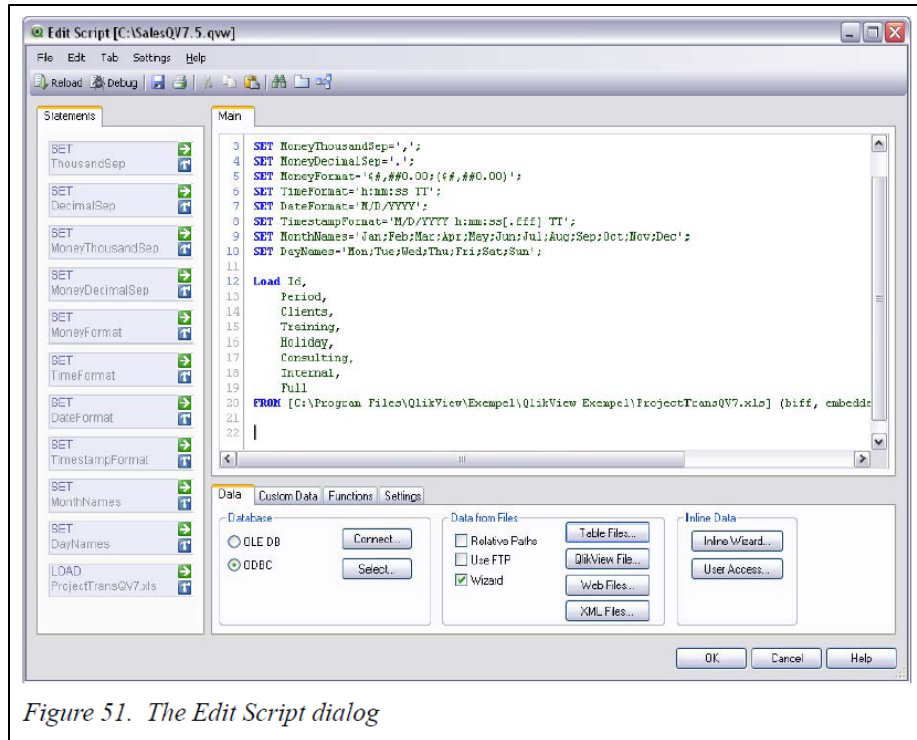


Figure 51. The Edit Script dialog

EX1008, 0229.

314. The Edit Script dialog above can read data from different data sources. For example, when the user selects the “Table Files” option, the Open Local Files dialog box (depicted below) is opened. EX1008, 0249. When a file is selected and the open button is clicked, a load statement is generated and inserted in the Edit Script dialog. EX1008, 0249.

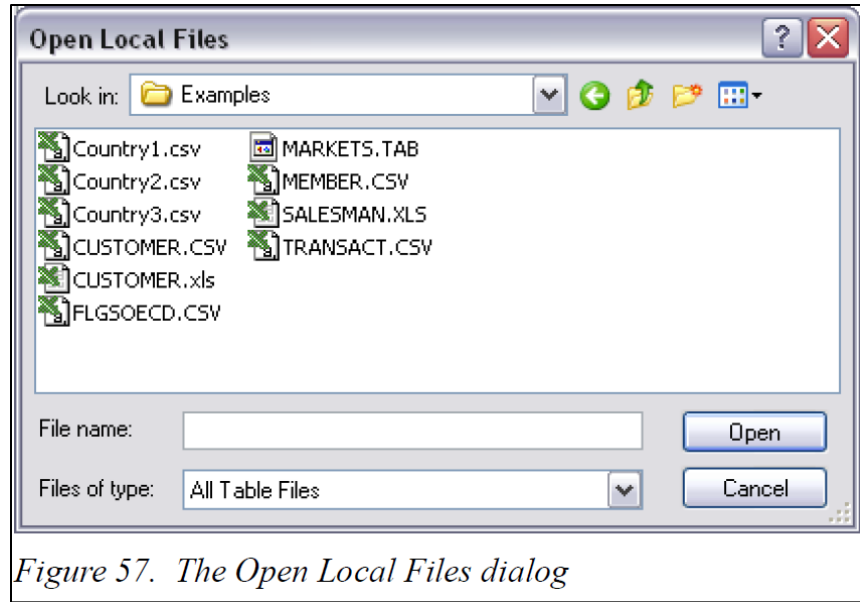
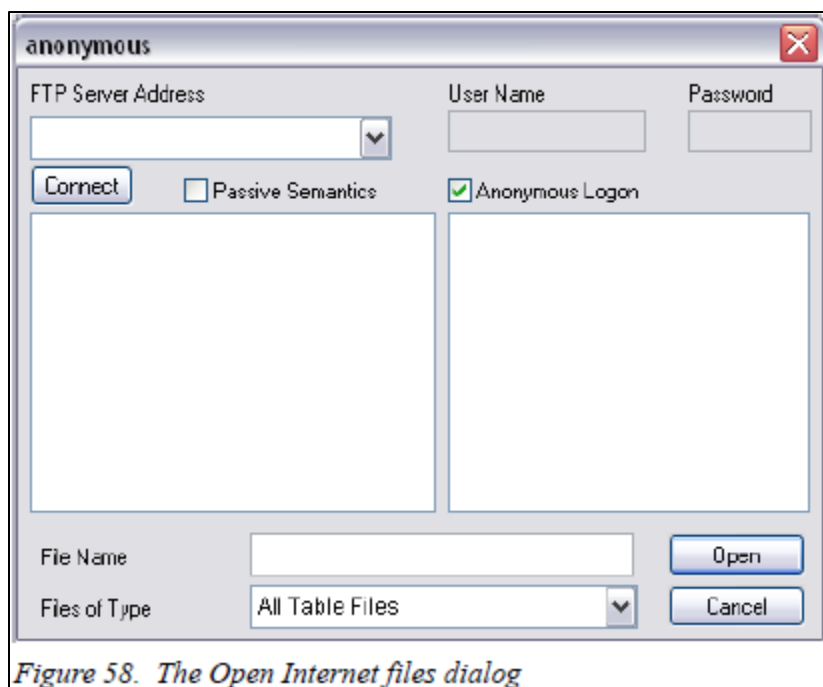


Figure 57. The Open Local Files dialog

EX1008, 0249.

315. Similarly, when the user selects Table Files or QlikView File buttons with the Use FTP box checked, the Open Internet Files dialog box (below) is shown. EX1008, 0250. Here, the user can connect to and load data from a server. EX1008, 0250. Similar to the operation above, a load statement is generated and inserted in the Edit Script dialog. EX1008, 0251.



EX1008, 0250.

316. Users can also load data from databases and other files. EX1008, 0229, 0242-251, 0255-57. For example, the Script Dialog can be used to connect to and load data via OLE DB and ODBC interfaces. EX1008, 0061-63, 0234-35. OLE DB (Object Linking and Embedding, Database) and ODBC (Open Database Connectivity) were known Microsoft-based APIs to access databases.

317. As a POSA would have understood, the script in the QlikView Manuals is part of the UI because users interface with it within the QlikView environment. For at least these reasons, the QlikView Manuals disclose [1D].

6. [1E] “receive a selection of the chart template via the user interface”

318. Claim 1 recites a processor that “receive[s] a selection of the chart template via the user interface.” The QlikView Manuals disclose this limitation.

319. As I explained for [1D], Section XIII.B.5, a user creates a new sheet object (e.g., a chart) via the UI. When creating a new chart via the UI, the processor receives a selection of the chart template. Shown below, when creating a new chart, the Chart Wizard’s layout tab has options for the processor to receive a selection of a chart theme/template from the Theme Maker Wizard. EX1008, 0942.

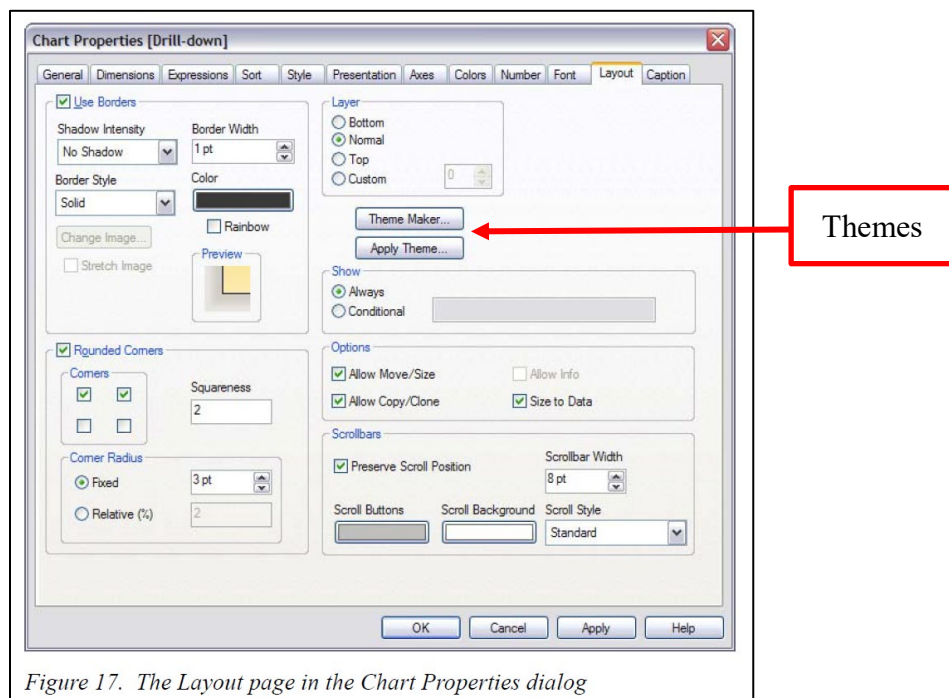


Figure 17. The Layout page in the Chart Properties dialog

EX1008, 0939 (annotated).

320. The QlikView Manuals explain the five steps for selecting an existing theme or creating a new theme and thereafter selecting it. EX1008, 0795-0805.

First, the user selects a theme file at the dialog below. The user can create a new theme or modify an existing theme. EX1008, 0799. If the user elects to modify an existing theme, they can select an existing theme from the default theme folder, or browse the other locations for a theme file to modify. EX1008, 0799.

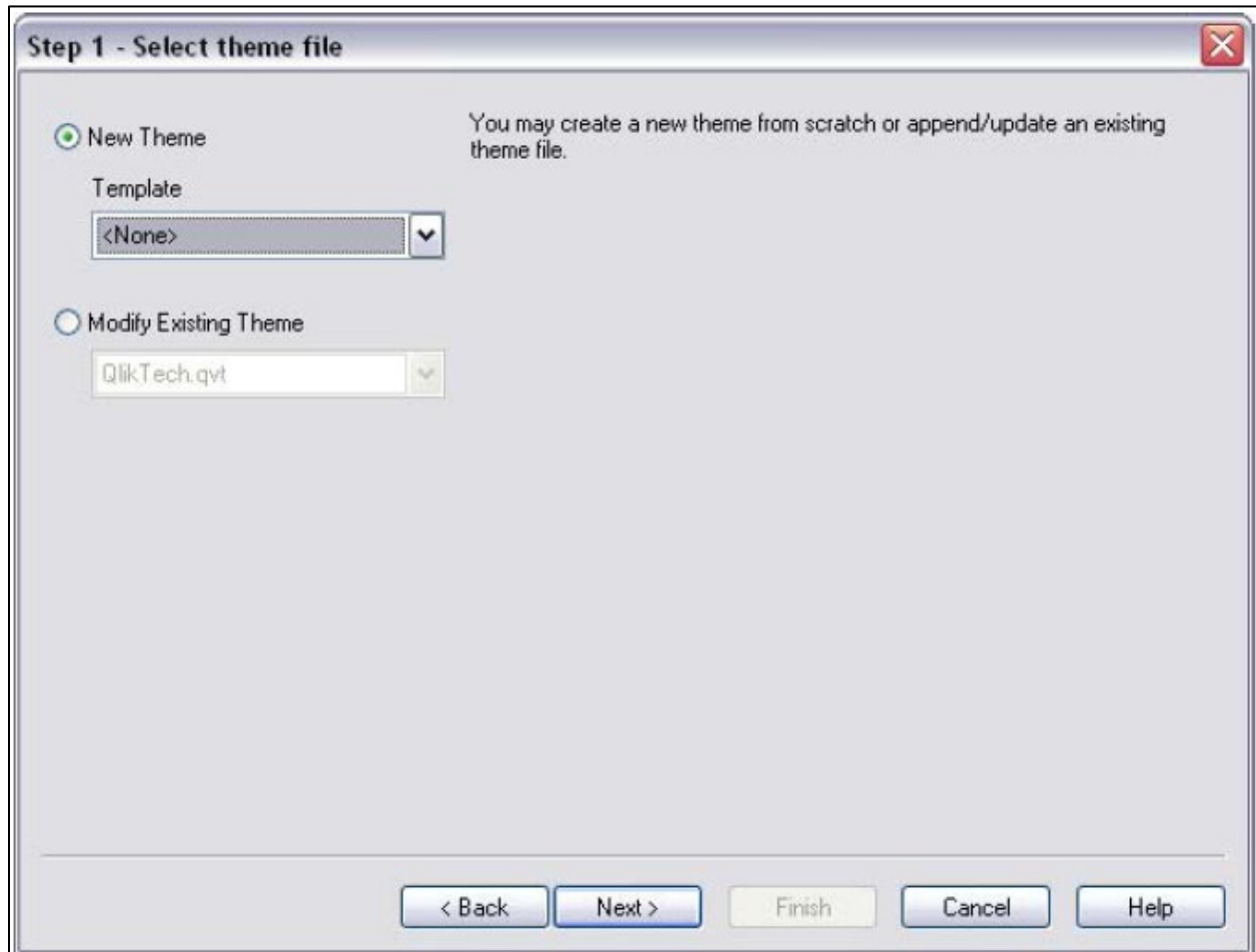


Figure 82. The first page of the Theme Maker Wizard

EX1008, 0799.

321. Next, the user can select a source object. Theme properties are extracted from the selected source object. EX1008, 0800. For example, as shown below, the user may select a slider object for property extraction.

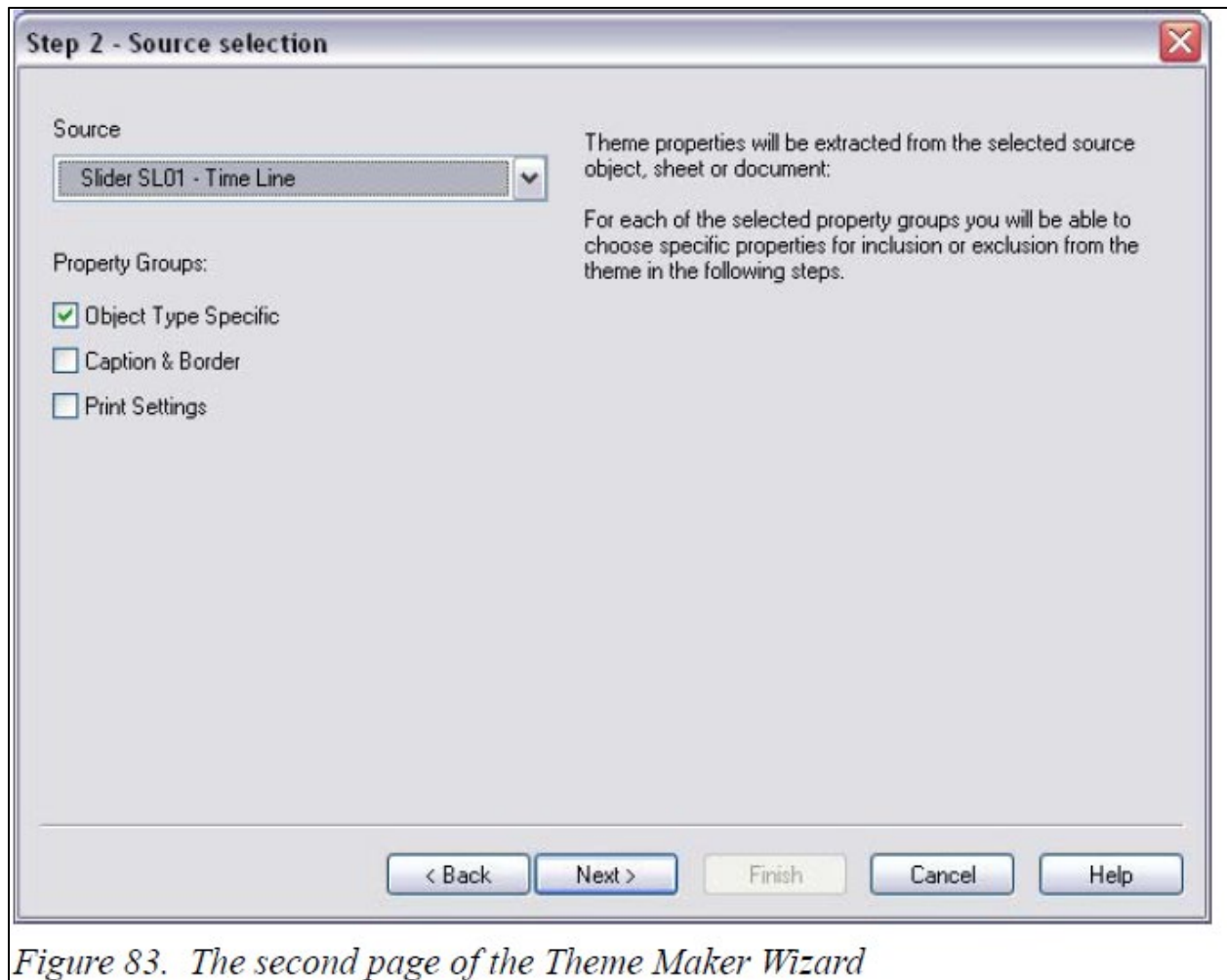
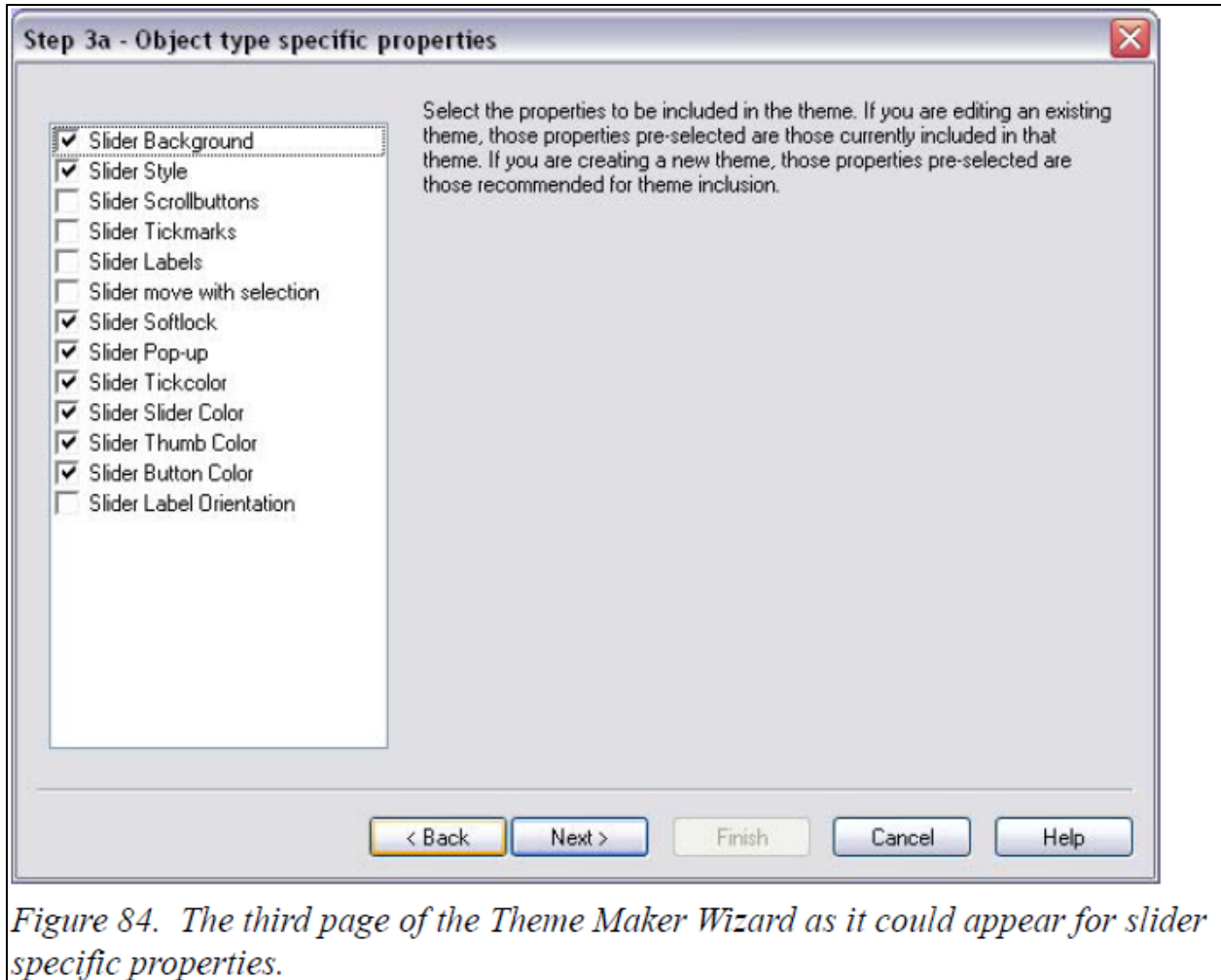


Figure 83. The second page of the Theme Maker Wizard

EX1008, 0800.

322. Subsequently, the user selects properties from the selected source object to include within the theme. EX1008, 0802. For example, the user can select to include the slider's background, style, and pop-up properties within the theme.

EX1008, 0802



EX1008, 0802.

323. Next, the user applies the extracted object properties. EX1008, 0803.

The user can apply the extract properties to sections and sub-sections of the theme.

EX1008, 0803. For example, the user can apply the extracted Caption & Border properties to List Boxes in the new theme. EX1008, 0803.

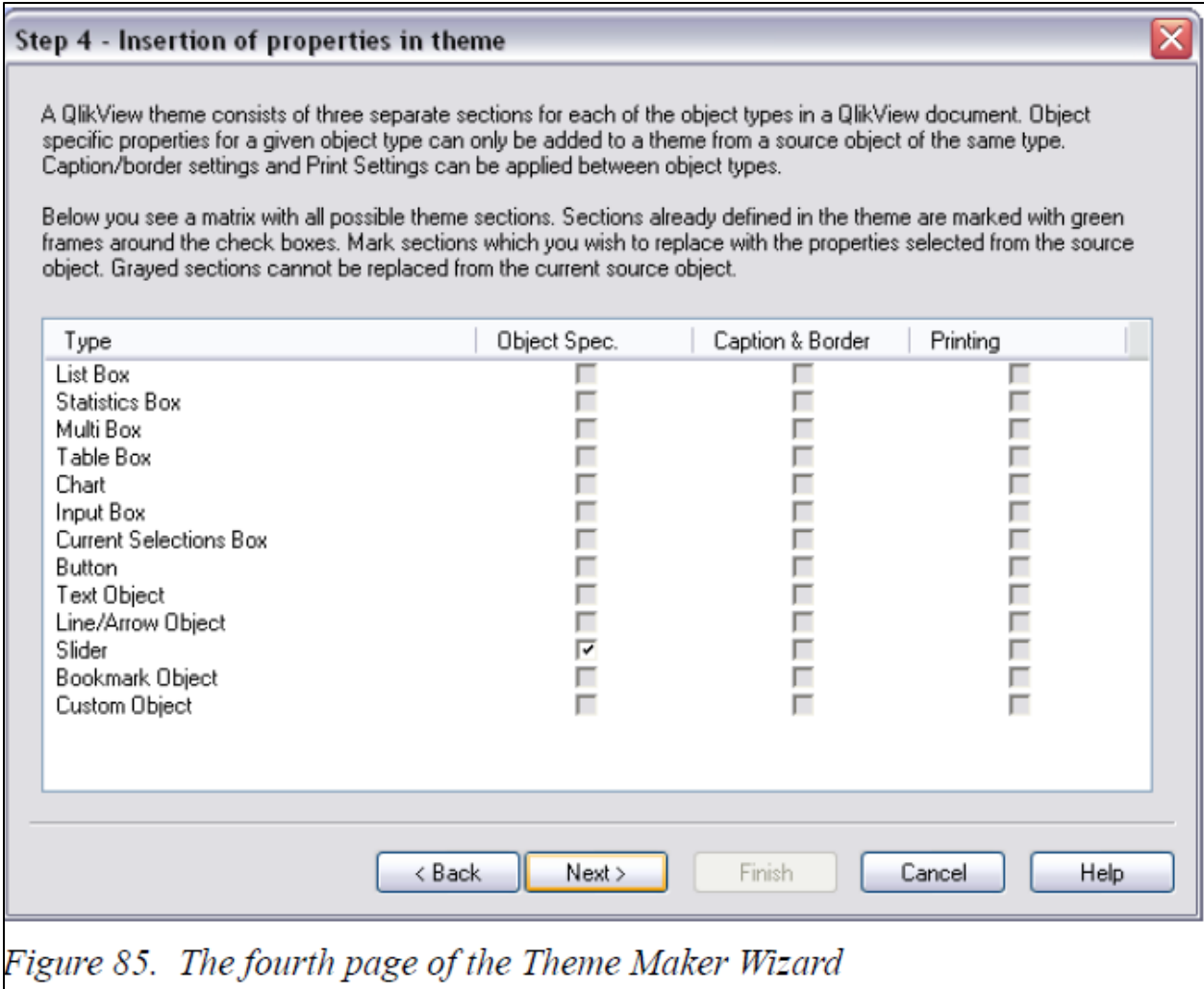
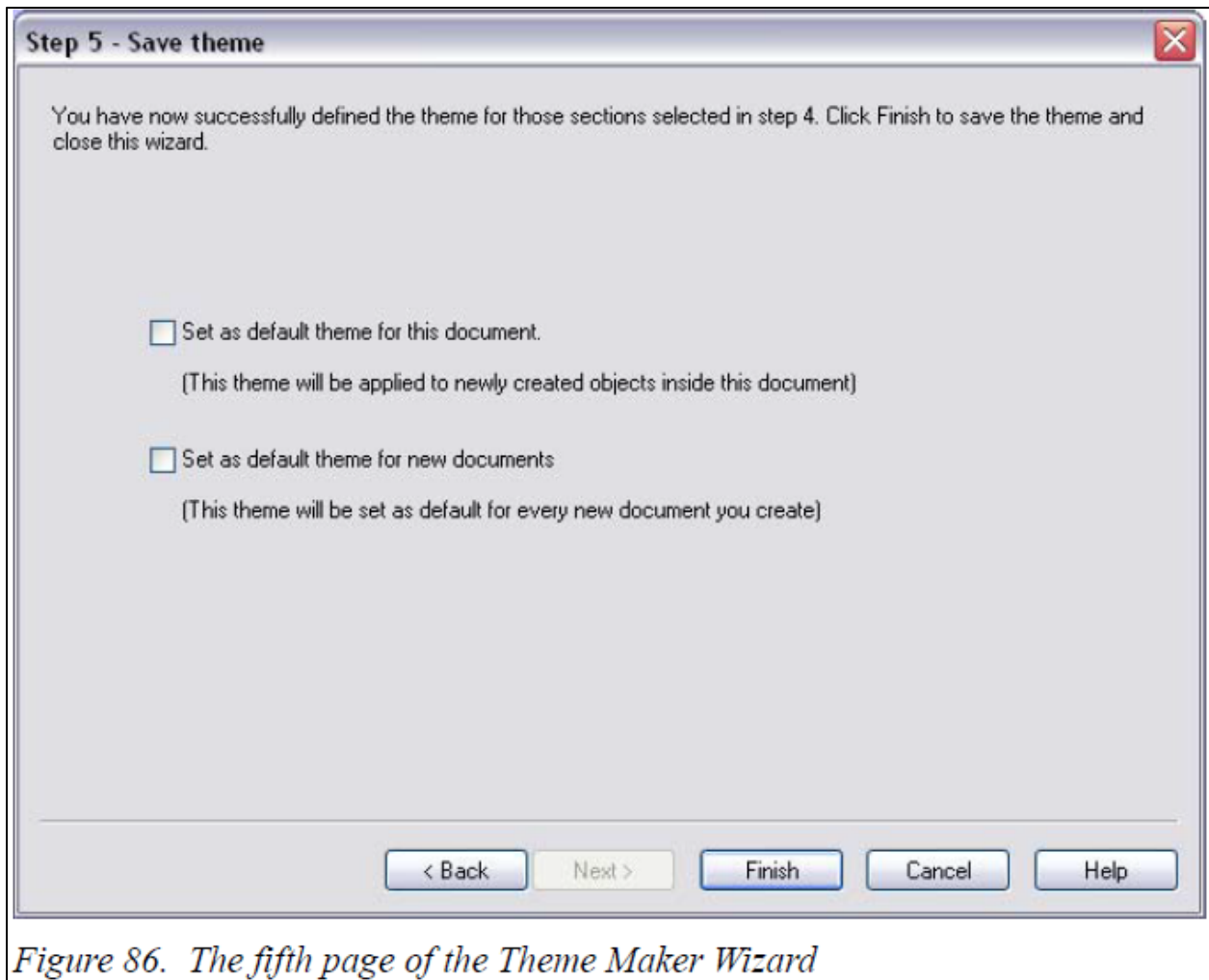


Figure 85. The fourth page of the Theme Maker Wizard

EX1008, 0803.

324. The user then saves the theme. EX1008, 0804. As part of saving the theme, the user selects whether to apply the theme to the current document or all new documents. EX1008, 0804.



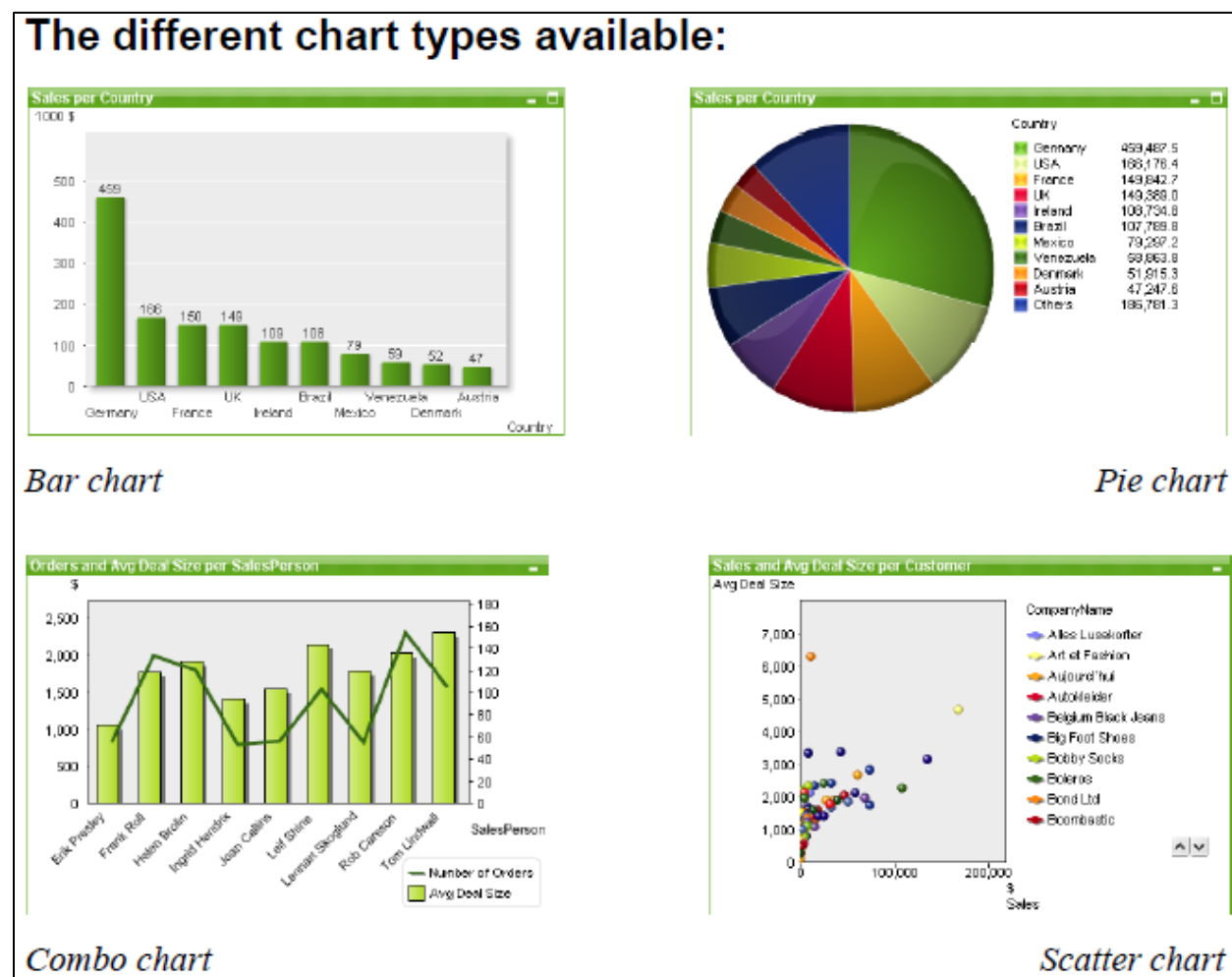
EX1008, 0804.

325. For at least these reasons, the QlikView Manuals disclose [1E].

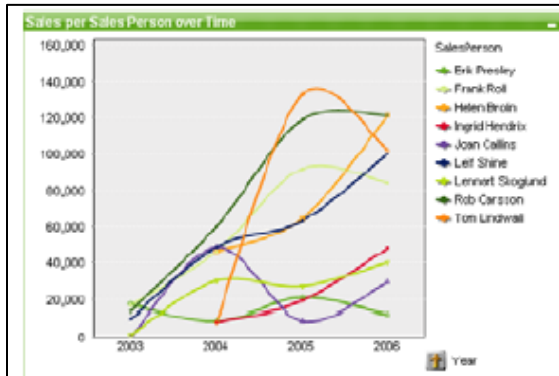
7. **[1F] “generate an interactive chart including the chart data and the chart template”**

326. Claim 1 recites a processor that “*generate[s] an interactive chart including the chart data and the chart template.*” The QlikView Manuals disclose this limitation.

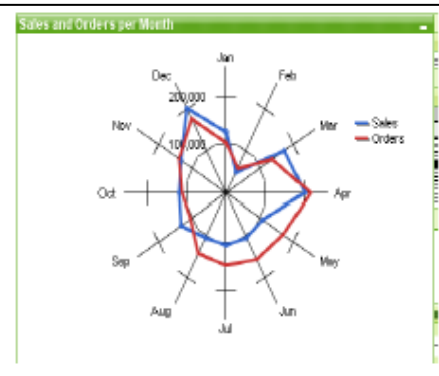
327. As I explained for [1B], Section XIII.B.3, the QlikView Manuals disclose a processor that performs the methods within a QlikView environment. After the processor receives the chart data and the chart template, *see* [1D], [1E], the processor running QlikView “*generate[s] an interactive chart including the chart data and the chart template.*” In QlikView, chart types include a “bar chart, line chart, combo chart, scatter chart, pie chart, radar chart, grid chart, block chart, gauge, funnel chart, pivot table and straight table.” EX1008, 0665; EX1008, 0883-85, 0887-89. Example generated charts are depicted below:



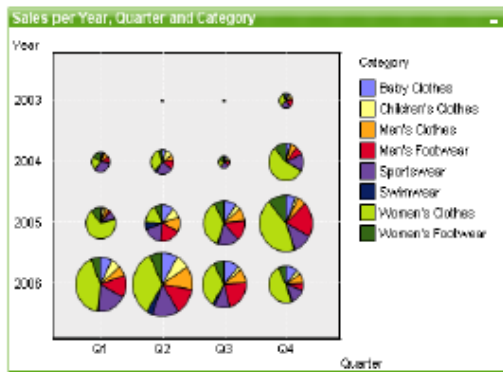
EX1008, 0887.



Line chart



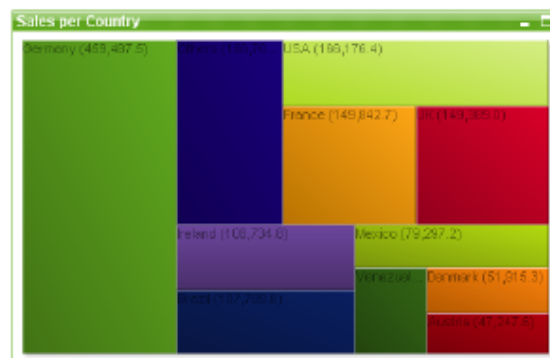
Radar chart



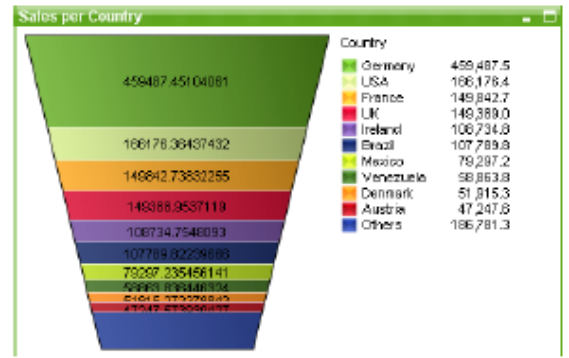
Grid chart



Gauge chart



Block chart



Funnel chart

EX1008, 0888.

Sales per CategoryName			
CategoryName	ProductName	Sales	Quantity
Men's Clothes	Alles Lussakotta	\$30,126.55	1057
	Bow tie	\$9,534.57	1315
	Desperado Jeans	\$18,340.88	706
	Lunin Jeansshorts	\$14,800.64	626
	M2 Trousers	\$17,944.48	1067
	Q-Man Underwear	\$1,849.67	296
	Roseli Bermuda Shorts	\$10,947.25	1367
	Samba Soccer Socks	\$9,941.14	1175
	US-Master Jeans	\$21,764.94	817
	Total	\$130,050.42	8660
Women's Clothes	Chantel Shirt	\$7,504.70	368
	Haier Dress	\$361,096.85	981
	Jack Flash Dress	\$42,636.00	722
	Lancoste Shirt	\$4,433.35	246
	Le Baby Dress	\$47,571.88	623
	Minni Palai	\$10,472.71	164
	Oxolaba Skin Jackets	\$42,258.78	601
	Oyaki Kimono	\$9,084.42	806

Pivot table

Sales per CategoryName			
CategoryName	ProductName	Sales	Quantity
Men's Clothes	Alles Lussakotta	\$30,126.55	1057
Men's Clothes	Bow tie	\$9,534.57	1315
Men's Clothes	Desperado Jeans	\$18,340.88	706
Men's Clothes	Lunin Jeansshorts	\$14,800.64	626
Men's Clothes	M2 Trousers	\$17,944.48	1067
Men's Clothes	Q-Man Underwear	\$1,849.67	296
Men's Clothes	Roseli Bermuda Shorts	\$10,947.25	1367
Men's Clothes	Samba Soccer Socks	\$9,941.14	1175
Men's Clothes	US-Master Jeans	\$21,764.94	817
Women's Clothes	Chantel Shirt	\$7,504.70	368
Women's Clothes	Haier Dress	\$361,096.85	981
Women's Clothes	Jack Flash Dress	\$42,636.00	722
Women's Clothes	Lancoste Shirt	\$4,433.35	246
Women's Clothes	Le Baby Dress	\$47,571.88	623
Women's Clothes	Minni Palai	\$10,472.71	164
Women's Clothes	Oxolaba Skin Jackets	\$42,258.78	601
Women's Clothes	Oyaki Kimono	\$9,084.42	806

Straight table

EX1008, 0889, 0883 (pivot tables and straight tables are charts).

328. The charts are highly customizable. For example, when generating a bar chart, the user can customize aspects of the chart such as its dimensions, animations, expressions, sort capabilities, style, presentation, legend, reference lines, text, axes, colors, numbers, fonts, layout, and captions. EX1008, 0890-0947; *see also* EX1008, 0948-959 (line chart customization), 0960-971 (combo chart customization), 0972-982 (radar chart customization), 0984-1000 (scatter chart customization), 1002-012 (grid chart customization), 1014-022 (pie chart customization), 1024-032 (funnel chart customization), 1034-044 (block chart customization), 1046-057 (gauge chart customization), 1058-081 (pivot table chart customization), 1082-093 (straight table chart customization).

329. The user can also choose different chart styles. EX1008, 0915, 0952. The styles for bar charts are shown below.

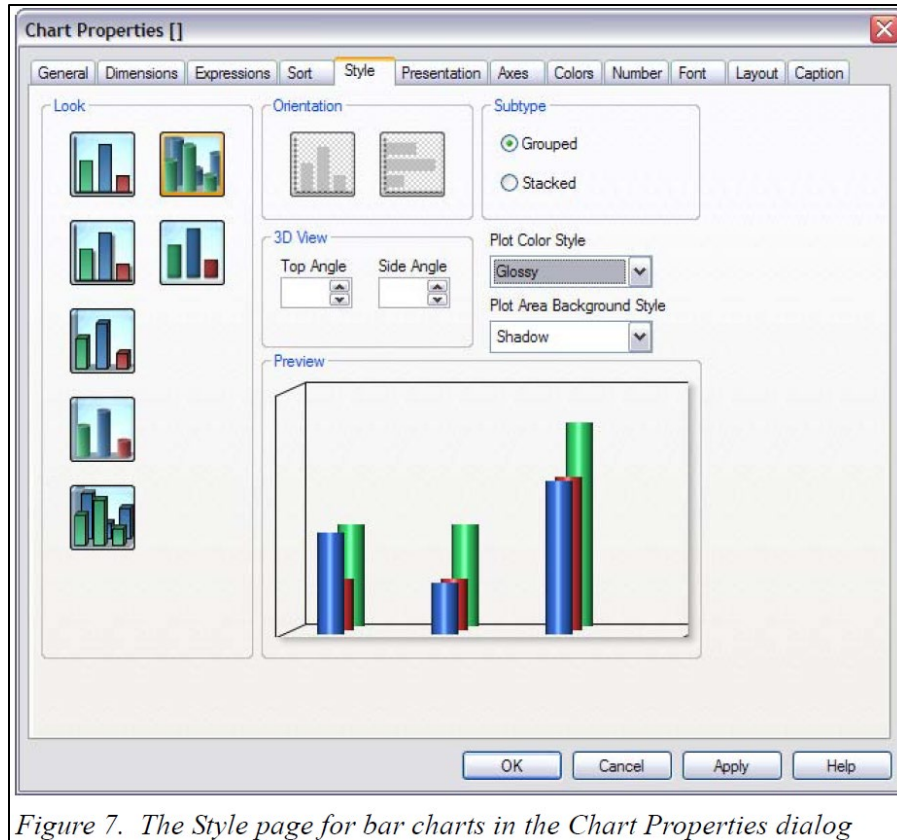


Figure 7. The Style page for bar charts in the Chart Properties dialog

EX1008, 0915.

330. After a user customizes the chart options, the processor generates the chart once the user selects ok, as shown below. EX1008, 0951.

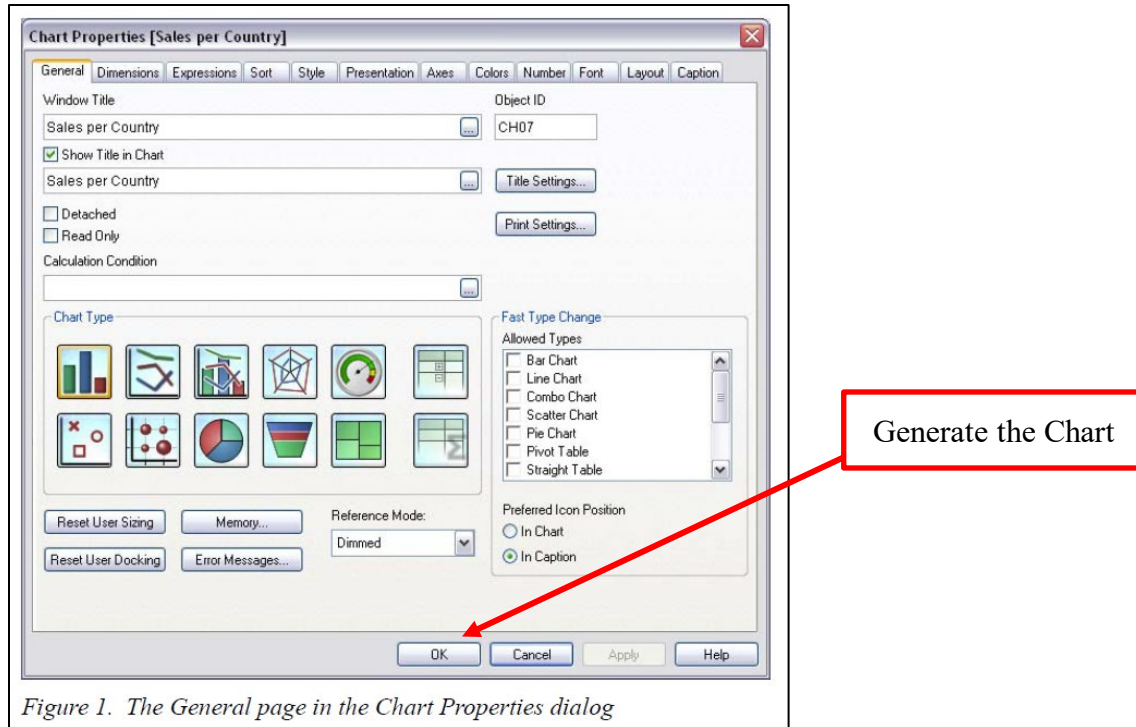
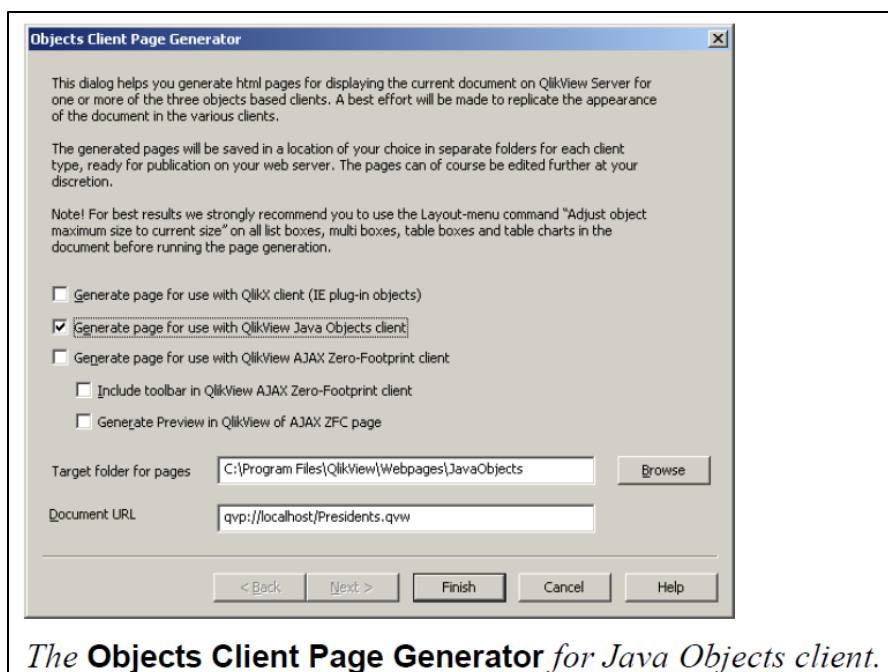


Figure 1. The General page in the Chart Properties dialog

EX1008, 0882, 0085, 0880, 0890, 1104-114 (the Quick Chart Wizard).

331. Additionally, the processor “*generate[s] an interactive chart including the chart data and the chart template*” when using the “Objects Client Page Generator.” QlikView’s Objects Client Page Generator (below) generates a page having sheet objects (e.g., charts). EX1009, 0134. As I explained above, these charts include the chart data and the chart template.



EX1009, 0134; *see also* EX1008, 0807.

332. QlikView’s generated charts are “*interactive*” because QlikView’s generated charts are dynamic. EX1008, 0013; *see also* Section XIII.B.1 (element [1P]). “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.

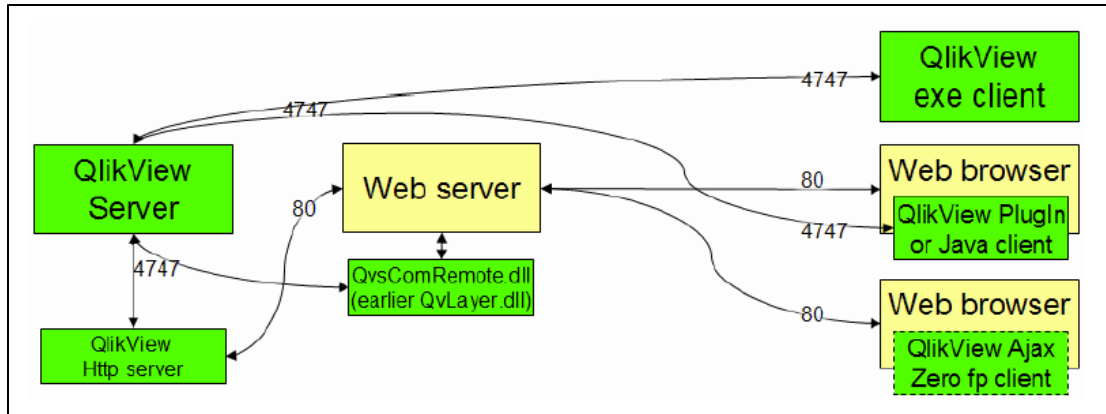
333. As a POSA would have understood, these interactive features are mutually or reciprocally active and provide real-time feedback to users to further modify the charts.

334. For at least these reasons, the QlikView Manuals disclose [1F].

8. **[1G] “a network interface that, in response to generation of the interactive chart, transmits the interactive chart to a second website for display without the execution of the instructions stored in the memory to generate the user interface of the first website”**

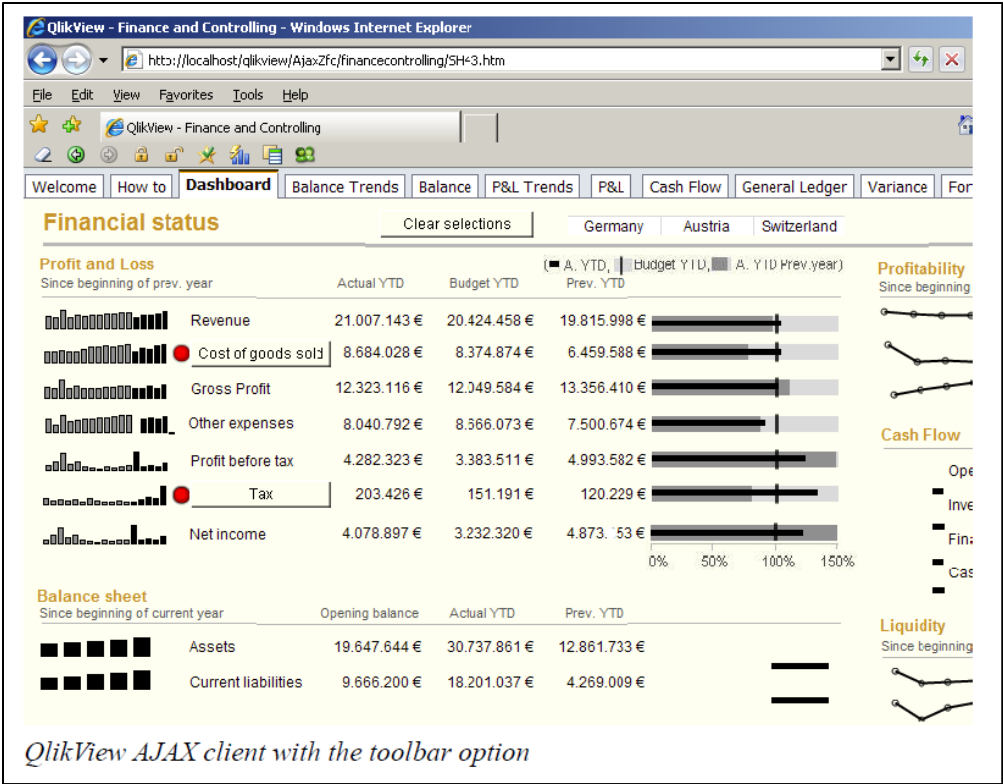
335. Claim 1 recites a processor that “*a network interface that, in response to generation of the interactive chart, transmits the interactive chart to a second website for display without the execution of the instructions stored in the memory to generate the user interface of the first website.*” The QlikView Manuals disclose this limitation.

336. The QlikView Manuals disclose “*a network interface*” because QlikView uses server-client interfaces within a “TCP/IP Network.” EX1009, 0025, 0098. Shown below, there are interfaces between the QlikView Server, Web Server, and Clients (“*network interface[s]*”). EX1009, 0095-0101. For example, the QlikView Server communicates with Windows clients via port 4747 and the HTTP Server communicates with AJAX clients via port 80. EX1009, 0098.



EX1009, 0098.

337. Depending on the client, the respective “*network interface*” “*transmits the interactive chart to a second website for display.*” EX1009, 0009, 0123-24, 0133-149, 155-66. For example, AJAX allows “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, 0156. Shown below, the transmitted pages have charts.



EX1009, 0155.

338. Similarly, in the Java Objects Client depicted below, each chart is implemented as a Java applet. EX1009, 0133. The Java applets are inserted within the html of the webpage (“the second website”). EX1009, 0133.

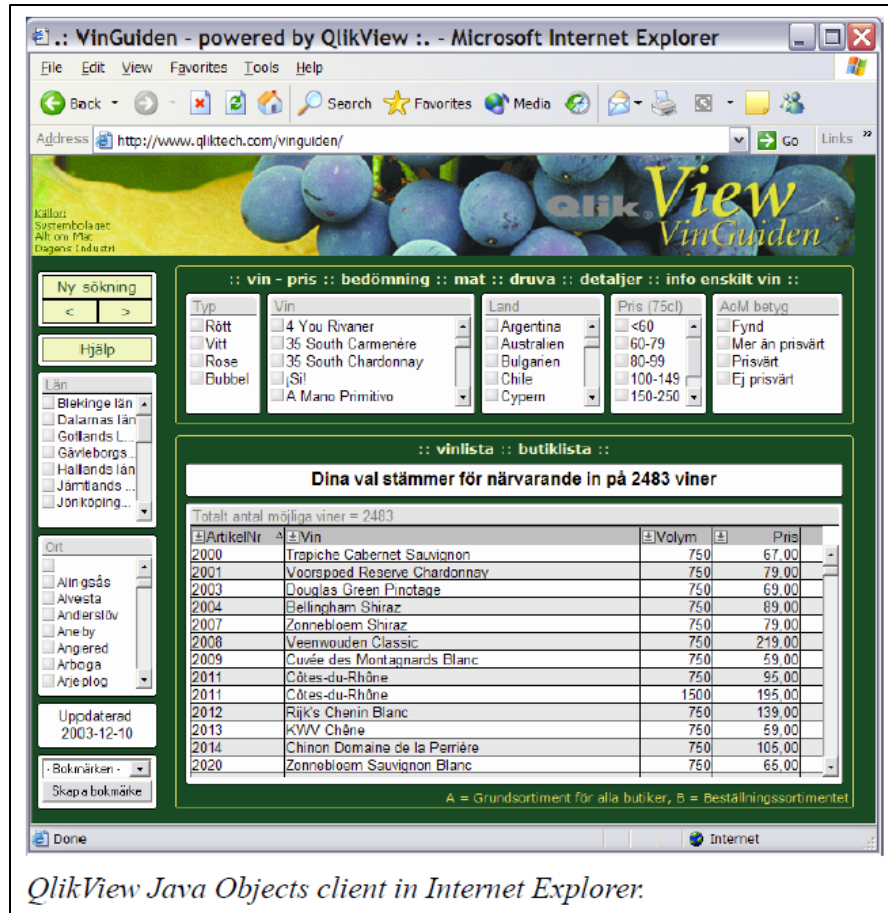


EX1009, 0133.

339. The second website is different than the first website discussed above for element [1C] 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 [1C]. Compare EX1009, 0133 (“http://www.qliktech.com/vinguiden”),

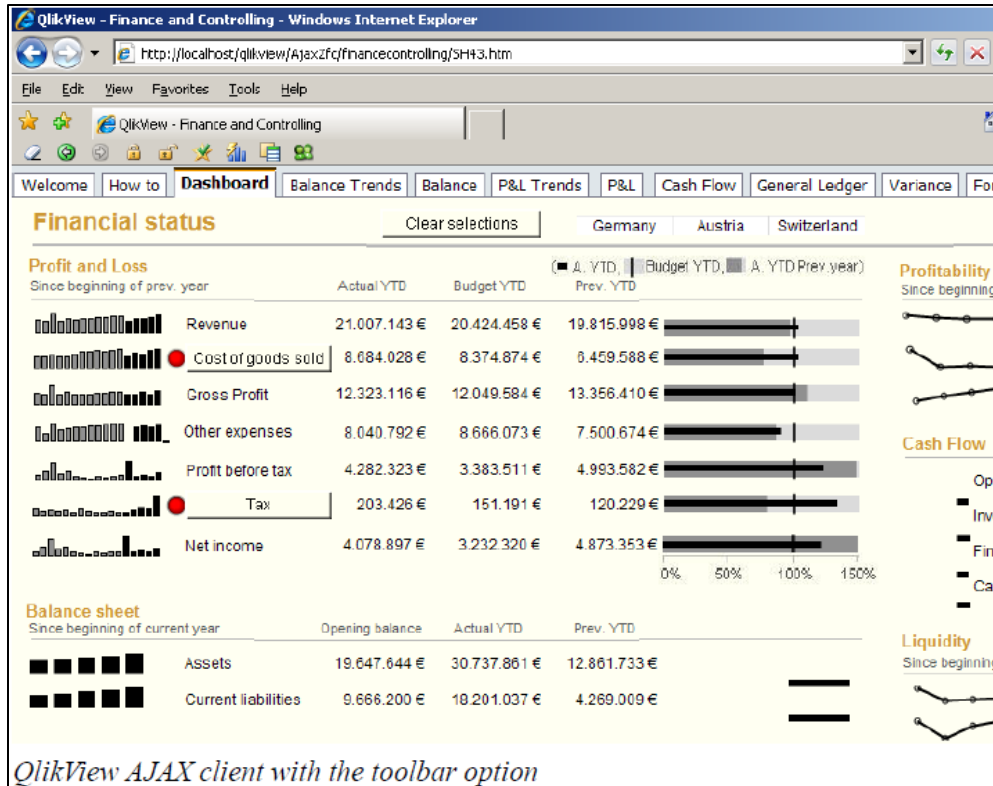
with EX1009, 0129, 0139 (“http://localhost/QvJava . . .” /
“http://localhost/QvPlugin . . .”).

340. When on the second website, the charts remain interactive. 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. The user can make selections by clicking options for the chart via the mouse. EX1008, 0881. The interactive charts include “drop-down select,” “slider object[s],” “buttons,” “interactive search feedback,” etc. EX1009, 0016-17; EX1008, 0035-36. “Java client users [can also] change certain column sizes in Multi Box and table/chart objects.” EX1009, 0057. The interactive chart 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.

341. 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 refreshing the page. EX1009, 0155-56.



EX1009, 0155.

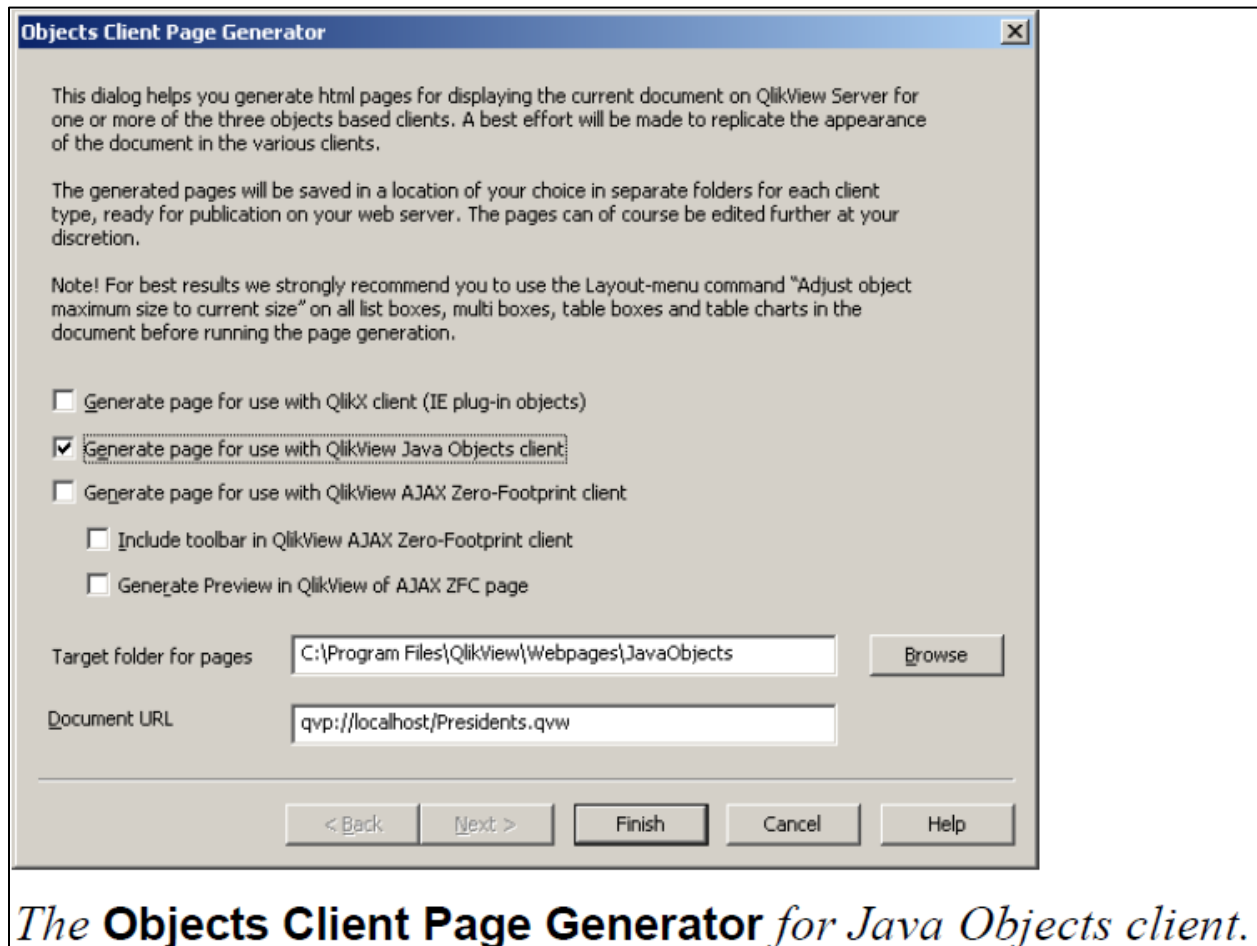
342. Accordingly, unlike the background art in the '595 patent, EX1001, 1:21-51, the QlikView charts *do* retain interactive features.

343. The transmission is "*in response to generation of the interactive chart*" because once the interactive chart and HTML page are generated, QlikView uses connector code to transmit the content to the second website for display automatically.

344. Shown below, the user can interface with the Objects Client Page Generator to "generate html pages for displaying the current document [e.g., chart] on [the] QlikView Server." EX1008, 0807. Using the Objects Client Page

Generator, the user specifies the “folder where the generated pages will be saved.”

EX1008, 0808. The user also specifies a Document URL indicating a “URL on the QlikView Server where the pages are to be run.” EX1008, 0808.



EX1009, 0134.

345. Connector code connects the source directory to the second website and transmits the content to the second website for display. Shown below, 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 identifier is referenced within the HTML to embed the chart within

the web page. EX1009, 0136. As a result, the “addObject” command and the chart identifier embed the chart in the webpage. EX1009, 0133.

```
<!--#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).

```

<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); EX1009, 0130, 0135 (Java applets).

346. The connector code examples above connect the source directory to the second website and transmit the content to the second website for display. With QlikView, this transmission occurs automatically when an interactive chart is generated because, for example, QlikView “automatically generate[s] AJAX ZFC pages” EX1009, 0016. Indeed, 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. *See* Section VI.C (overview of AJAX); *see also* EX1009, 0012-16, 0155-56; EX1008, 0020-21.

347. The QlikView Manuals also disclose that the network interface transmits the interactive chart to a second website “*without the execution of the*

instructions stored in the memory to generate the user interface of the first website.” Indeed, the QlikView Manuals disclose that a major theme of QlikView is “the possibility to change certain field values without running the script.” EX1008, 0023.

348. As a POSA would have understood, publishing the interactive chart to a second website is independent of the generation interface (the Chart Wizard). The publishing relies on self-contained applications that display the interactive charts independently from the generation interface.

349. 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.

350. “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. Because the QlikView interactive charts execute as Java applets, restricted to the JRE, they execute on the second

website “*without the execution of the instructions stored in the memory to generate the user interface of the first website.*”

351. For at least these reasons, the QlikView Manuals disclose [1G].

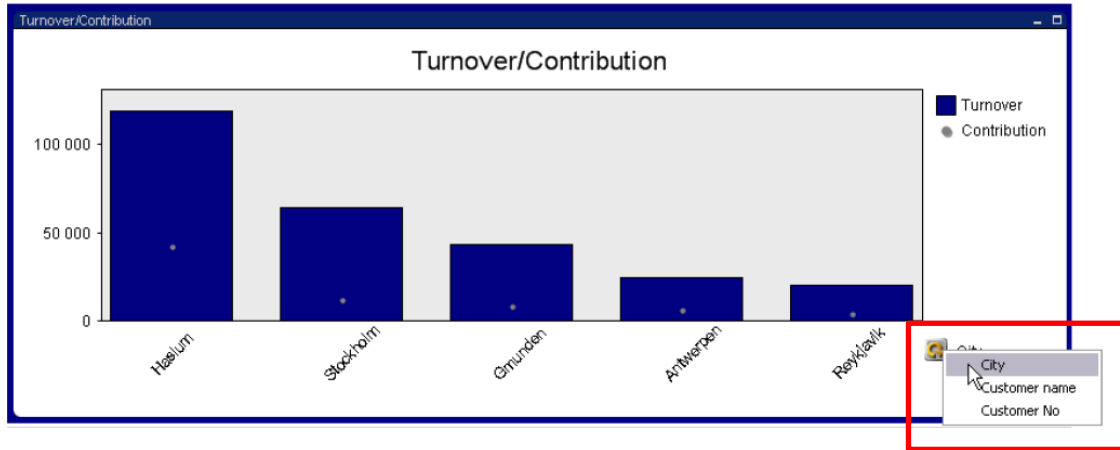
9. [1H] “wherein the second website receives an input from a second user and updates the interactive chart after the second website receives the input.”

352. Claim 1 recites “*wherein the second website receives an input from a second user and updates the interactive chart after the second website receives the input.*” The QlikView Manuals disclose this limitation.

353. The QlikView Manuals disclose “*the second website receiv[ing] an input from a second user*” because a second user (e.g., an end user), EX1008, 1359, provides input (e.g., mouse clicks) to update the chart. EX1008, 0013, 0015, 0881; EX1009, 0133, 0155-56, 0204. For example, “[s]elections in field data can be made directly in all of QlikView’s charts and tables by means of clicking or painting with the mouse.” EX1008, 0881. For instance, selections (e.g., interactions) are made in the plot area, chart legend, dimension axes, and expression axes. EX1008, 0154. The user can also interact with chart components (e.g., title, legend, text) to move and resize them. EX1008, 0669.

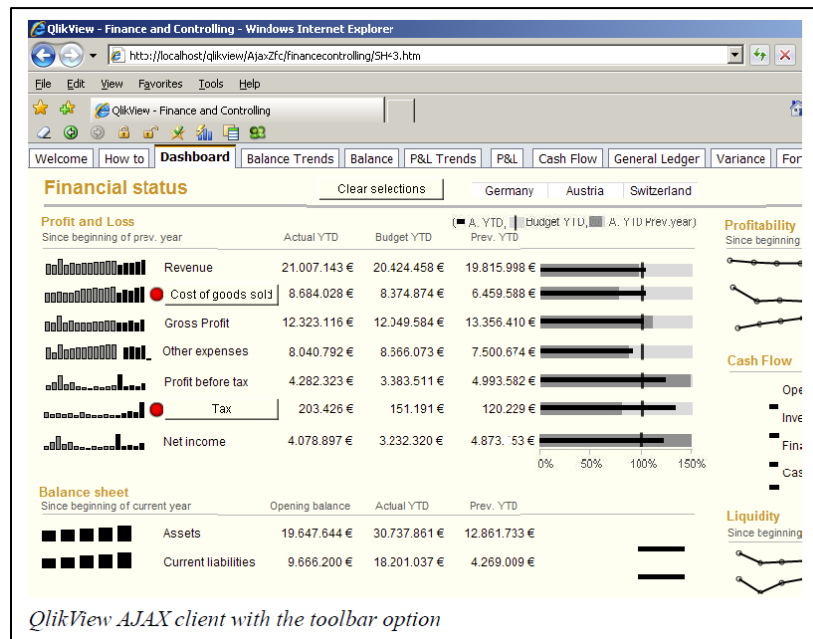
354. The second website receives the input from the second user through various interactive features such as “drop-down select,” “slider object[s],” “buttons,” and “interactive search.” EX1009, 0016-17, 0019. QlikView also

supports a drill down feature where a user selects a cycle icon to change the chart dimension field (e.g., “City,” “Customer name,” “Customer No”). EX1008, 1307.



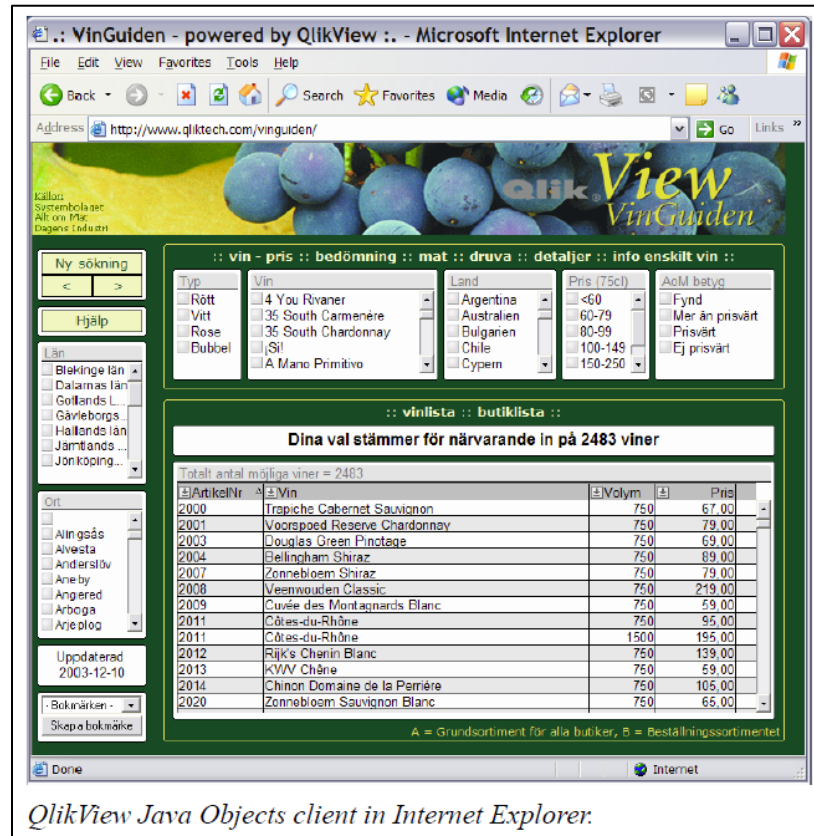
EX1008, 1307 (annotated).

355. Additionally, the AJAX Zero-Footprint Client shown below supports sorting data, editing input fields, searching table expressions, fast type chart change, and right-click context menus. EX1009, 0012, 0018-19.



EX1009, 0155.

356. In the Java Object Client shown below, the user can sort data and use scrollbars to navigate the data within the chart. EX1009, 0133.



EX1009, 0133.

357. The second website “*updates the interactive chart after the second website receives the input.*” For example, “[o]ne 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.” EX1009, 0012, 0014, 0155-56.

358. For at least these reasons, the QlikView Manuals disclose [1H].

C. Independent Claims 7 and 13

359. The QlikView Manuals teach or suggest every element in claims 7 and 13. The claims are nearly identical, so I will discuss both collectively.

1. **[7P] “A method for creating interactive charts” / [13P] “A non-transitory computer readable storage medium having embodied thereon a program executable by a processor for creating interactive charts”**

360. Claim 7 recites “[a] *method for creating interactive charts.*” Claim 13 recites “[a] *non-transitory computer readable storage medium having embodied thereon a program executable by a processor for creating interactive charts.*” To the extent the preambles are determined to be limiting, the QlikView Manuals disclose the preambles of claims 7 and 13.

361. As I explained in Section XIII.B.1, the QlikView Manuals disclose a system for creating interactive charts. This system performs the following method. And “[a] *non-transitory computer readable storage medium ha[s] embodied thereon a program executable by a processor for creating interactive charts*” for the same reasons the QlikView Manuals disclose [1B] (Section XIII.B.3).

2. **[7A]/[13A] “generating a user interface on a first website”**

362. Claims 7 and 13 both recite “*generating a user interface on a first website.*” The QlikView Manuals disclose this limitation for the same reasons they disclose [1C] (Section XIII.B.4).

3. [7B]/[13B] “receiving a selection of a chart template via the user interface”

363. Claims 7 and 13 both recite “*receiving a selection of a chart template via the user interface.*” The QlikView Manuals disclose this limitation for the same reasons they disclose [1E] (Section XIII.B.6).

4. [7C]/[13C] “receiving entry of chart data via the user interface”

364. Claims 7 and 13 both recite “*receiving entry of chart data via the user interface.*” The QlikView Manuals disclose this limitation for the same reasons they disclose [1D] (Section XIII.B.5).

5. [7D]/[13D] “generating an interactive chart including the chart data and the chart template”

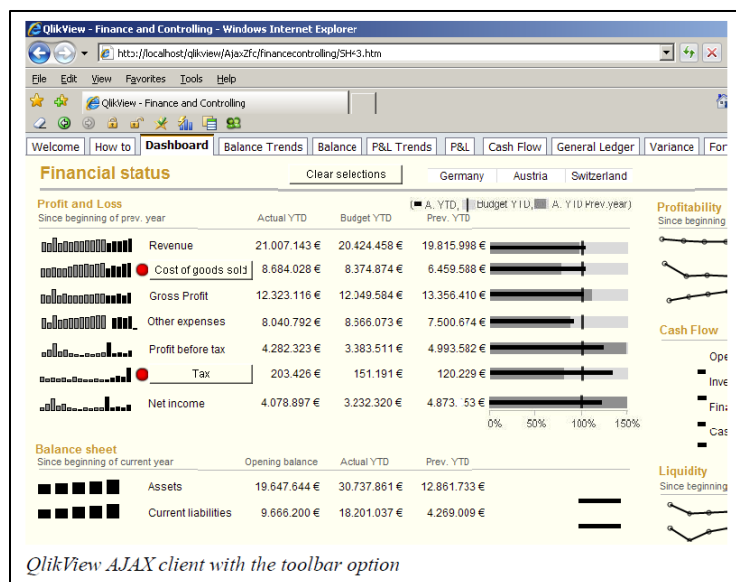
365. Claims 7 and 13 both recite “*generating an interactive chart including the chart data and the chart template.*” The QlikView Manuals disclose this limitation for the same reasons they disclose [1F] (Section XIII.B.7).

6. [7E]/[13E] “transmitting the interactive chart to a second website for display in response to generation of the interactive chart”

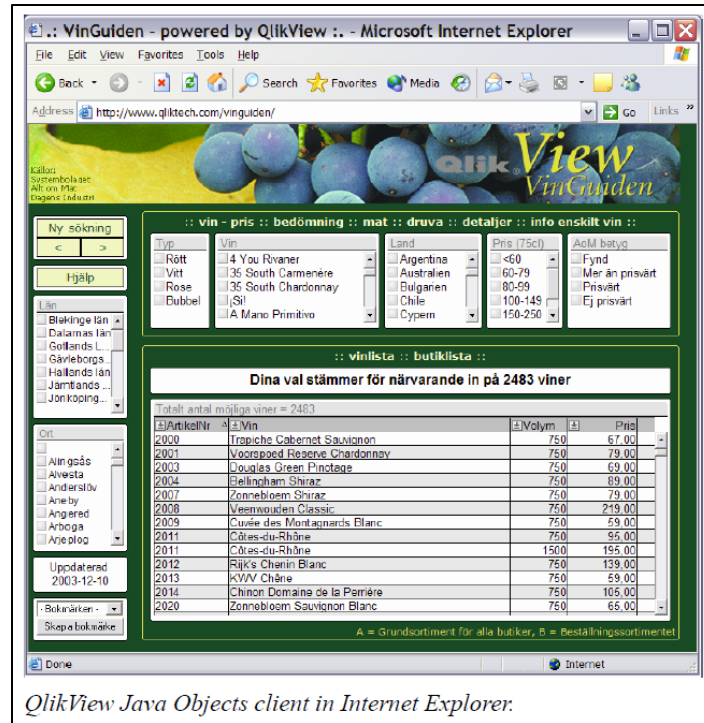
366. Claims 7 and 13 both recite “*transmitting the interactive chart to a second website for display in response to generation of the interactive chart.*” The QlikView Manuals disclose this limitation for the same reasons they disclose [1G] (Section XIII.B.8).

7. [7F]/[13F] “wherein: the interactive chart is displayed via the second website without generating the user interface of the first website”

367. Claims 7 and 13 both recite “wherein: the interactive chart is displayed via the second website without generating the user interface of the first website.” As I explained above for [1G] (Section XIII.B.8), “the interactive chart is displayed via the second website.” Two examples of the second website are below.



EX1009, 0155.



EX1009, 0133.

368. The second website is independent of the first website's UI, so the interactive chart is displayed *"without generating the user interface of the first website."* As the QlikView Manuals explain, the second website is self-contained and independent from the first website's UI. For example, the second website can execute using a Java Objects Client where each chart is implemented as a Java applet in a webpage (*"the second website"*). EX1009, 0133.

Java programs which are **self-contained applications** written in Java as opposed to e.g. VB or VC++.

EX1009, 0199.

The JRE is a plug-in to an internet browser which creates a "sandbox" for running applets written in Java inside the browser.

EX1009, 0199.

The fourth and final major theme of QlikView 8 is the introduction of the possibility to **change certain field values without running the script.**

EX1008, 0023.

369. As the QlikView Manuals explain, an end user can open, view, and interact with the chart on a web client without generating the first website's UI. For example, an end user with QlikView Analyzer can view and interact with the chart, independent of QlikView Professional/Developer, which enable users to generate new charts. EX1008, 0016. Additionally, as I explained for [1G], Section XIII.B.8, the second website is displayed on a different client using a different URL, which provides additional reasons why the second website is independent.

8. [7G]/[13G] “the second website receives an input from a second user”

370. Claims 7 and 13 both recite “*wherein . . . the second website receives an input from a second user.*” The QlikView Manuals disclose this limitation for the same reasons they disclose [1H] (Section XIII.B.9).

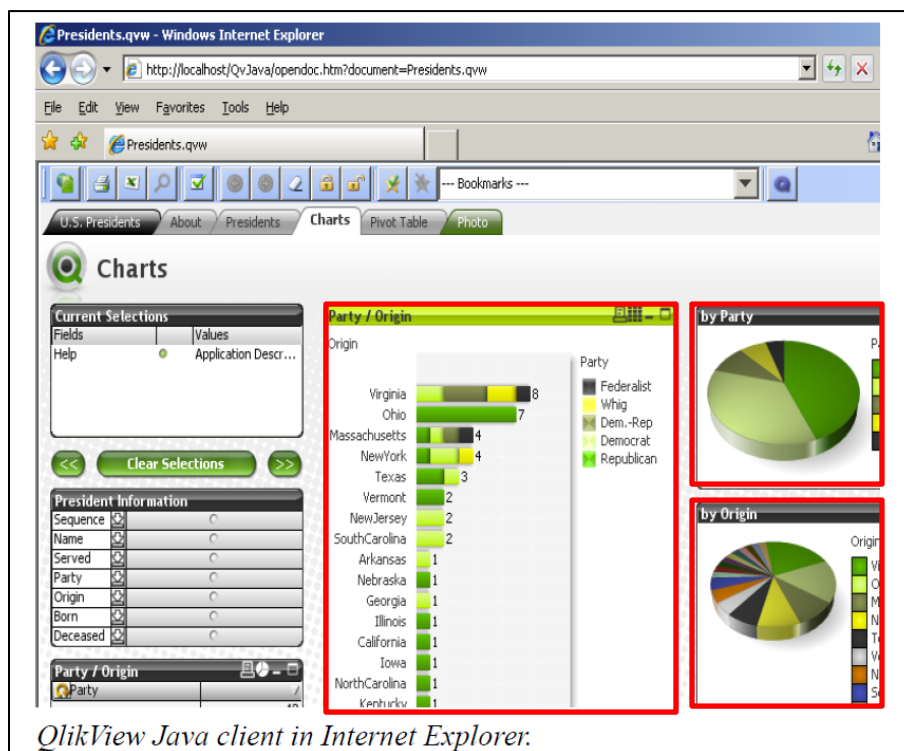
9. [7H]/[13H] “the interactive chart is updated after the second website receives the input.”

371. Claims 7 and 13 both recite “*wherein . . . the interactive chart is updated after the second website receives the input.*” The QlikView Manuals disclose this limitation for the same reasons they disclose [1H], Section XIII.B.9.

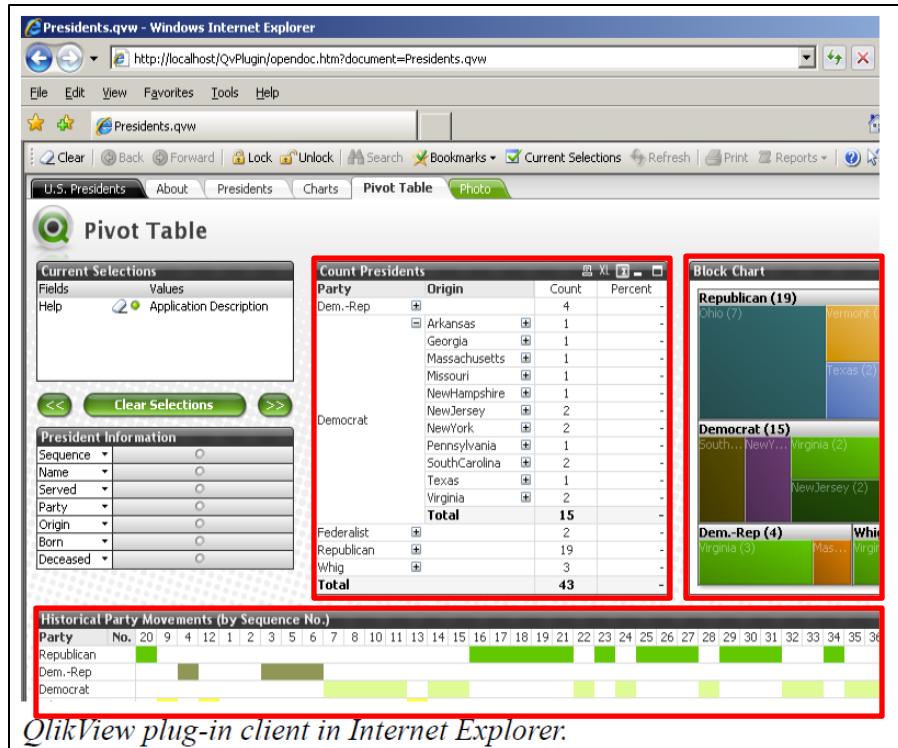
D. Claims 2, 8, and 14

372. Claims 2, 8, and 14 depend from claims 1, 7, and 13 respectively and all recite “*wherein the user interface includes a chart pane that displays the generated interactive chart.*” The QlikView Manuals disclose this limitation.

373. Shown below, the user interfaces identified for [1C] have chart panes that display the generated interactive charts. Other interfaces used to generate the interactive charts would likewise have chart panes to display the charts.



EX1009, 0129 (annotated).



EX1009, 0139 (annotated).

374. For at least these reasons, the QlikView Manuals disclose the additional limitation in claims 2, 8, and 14.

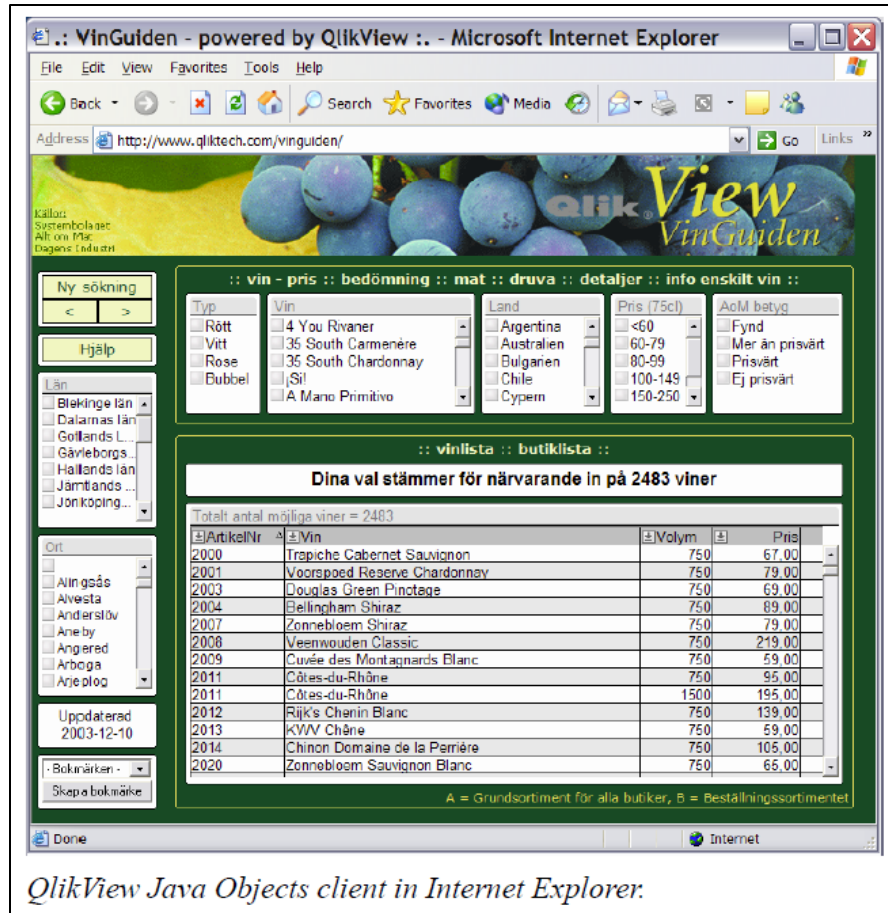
E. Claims 4, 10, and 16

375. Claims 4, 10, and 16 depend from claims 1, 7, and 13 respectively and all recite “*wherein the interactive chart retains one or more interactive features when displayed via the second website.*” The QlikView Manuals disclose this limitation.

376. As the QlikView Manuals explain, the charts retain interactive features when displayed via the second website. EX1009, 0133, 0155. 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. The user can make selections by clicking options for the chart via the mouse. EX1008, 0881. In the Java client, users can change column sizes within multi box tables and chart objects. EX1009, 0057.

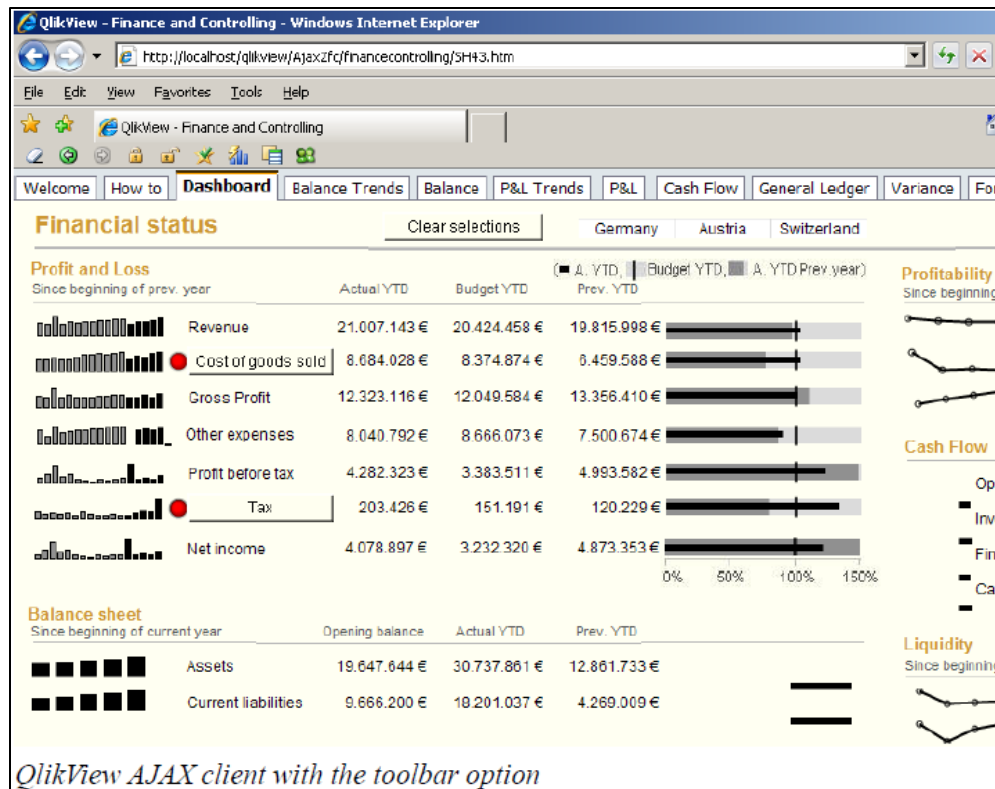
377. Additional interactive features include “drop-down select in tables,” “slider object[s],” “buttons,” “interactive search feedback,” etc. EX1009, 0016-17, 0019. The webpage further supports right click context menus and fast type change in charts. EX1009, 0018-19. As shown below, data can also be sorted and edited within a chart. EX1009, 0012, 0019, 0133. Additionally, the user can use scrollbars to navigate chart data. EX1009, 0133.



QlikView Java Objects client in Internet Explorer.

EX1009, 0133.

378. 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.

379. Accordingly, unlike the background art in the '595 patent, EX1001, 1:21-51, the QlikView charts *do* retain interactive features.

380. For at least these reasons, the QlikView Manuals disclose the additional limitation in claims 4, 10, and 16.

F. Claims 5, 11, and 17

381. Claims 5, 11, and 17 depend from claims 4, 10, and 16 respectively and all recite "*wherein the one or more interactive features include at least one of a zoom feature, a data mouse-over pop-up window, a data hide feature, a data share feature, an audio feature, a video feature, an animation feature, and a drill-*

down feature, or[/and] a data gliding feature.” The QlikView Manuals disclose this limitation.

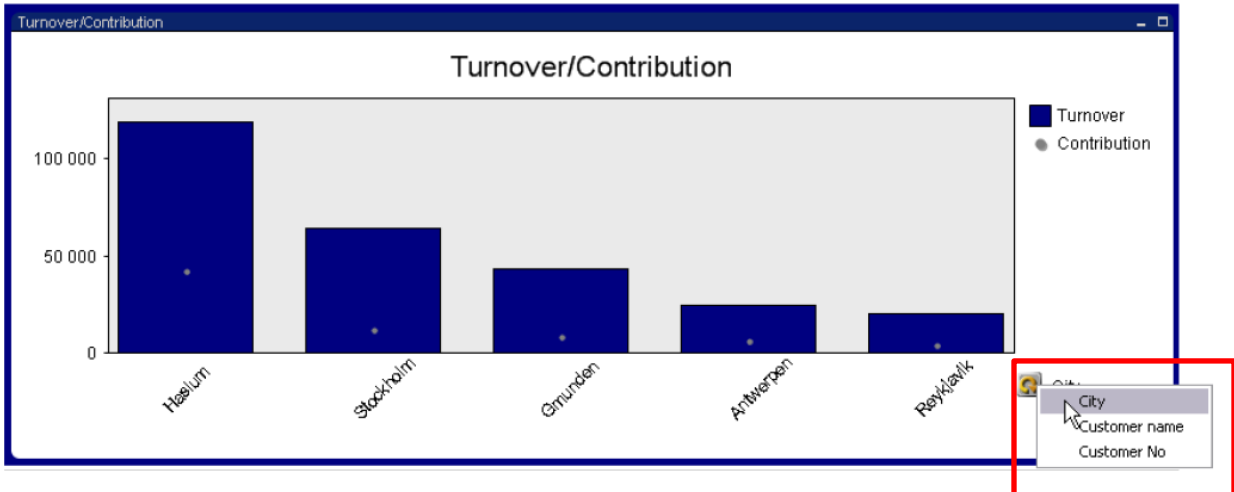
382. The QlikView Manuals disclose this limitation because they describe interactive features listed, including a data mouse-over pop-up window, EX1008, 0919, an animation feature, EX1008, 0895, 0899-0900, and a drill-down feature, EX1008, 1306-07.

383. For example, when an animation is added to a chart, an Animation Bar (shown below) is displayed at the bottom of the chart area. EX1008, 0899. The Animation Bar displays an interactive Play/Pause button. EX1008, 0899. The user can also interact with the progress bar to move between animation frames by clicking and dragging. EX1008, 0899.



EX1008, 0900.

384. Using the drill down feature illustrated below, the user can select a cycle icon to change the chart dimension field (e.g., “City,” “Customer name,” “Customer No”). EX1008, 1307.



EX1008, 1307 (annotated).

385. For at least these reasons, the QlikView Manuals disclose the additional limitation in claims 5, 11, and 17.

G. Claims 6, 12, and 18

386. Claims 6, 12, and 18 depend from claims 1, 1, and 13 respectively and all recite “*wherein the interactive chart is stored in [the] memory and is selectable by a subsequent user.*” The QlikView Manuals disclose this limitation.

387. The QlikView Manuals explain that documents and the sheet objects (e.g., charts) within documents are stored in the client’s local memory:

Document Folder: **Specify where documents to be accessed from any QlikView client are located.** This will typically reflect the default document location. . . . It will be located under the C:\ProgramData\QlikTech\QVS\Documents folder.

EX1009, 0059.

388. The QlikView Manuals also explain that “[o]bjects may be created in the usual manner but will be stored in a repository file, located in the same folder as the document on the server.” EX1009, 0013. This is the same local memory recited in [1B]. EX1008, 1326-28; EX1009, 0025-26.

389. The interactive chart is also “*selectable by a subsequent user*” because users can share their interactive charts with other users.

Types of **Objects Available for Sharing**: There are multiple objects available for user collaboration and sharing through QlikView Server.

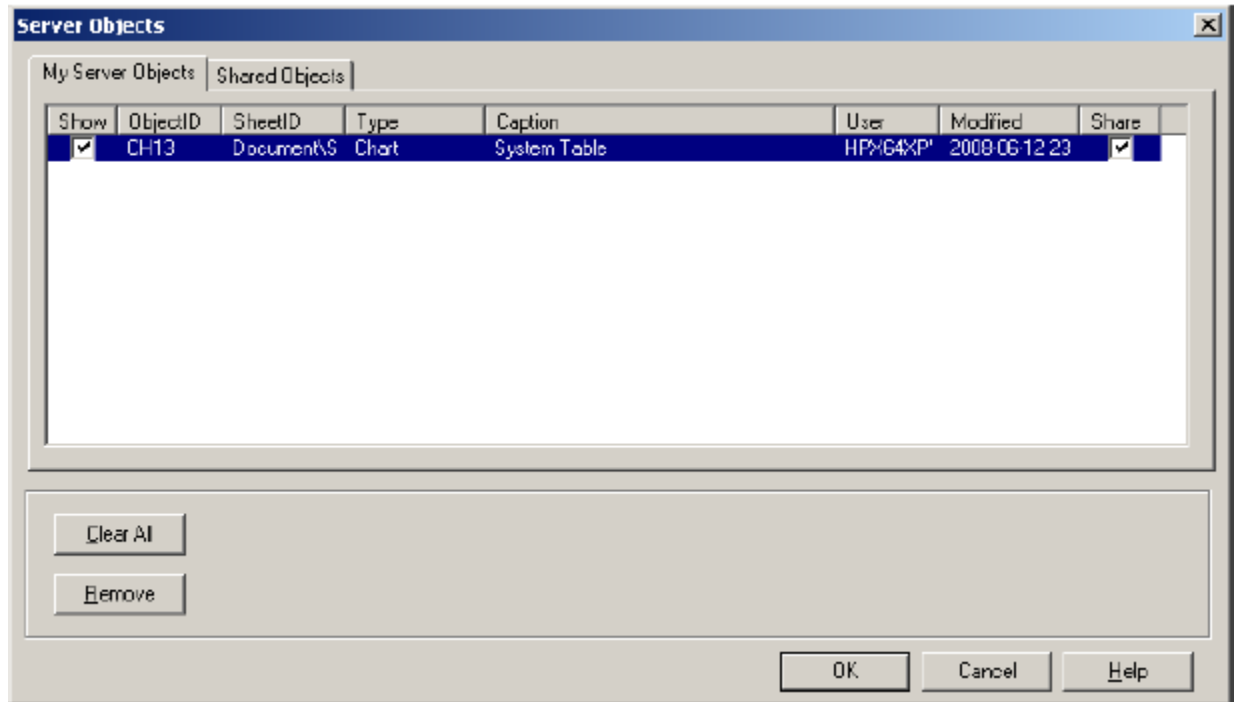
- Bookmarks
- Sheet objects, including **charts**
- Reports

Each of these objects may be defined as a user object, available to the authenticated user, regardless of access method or location, or a shared object, **available to all users** of the document through QlikView Server.

EX1009, 0113; *see also* EX1008, 0016, 0022, 0623-28, 0777-781, 1356-57;

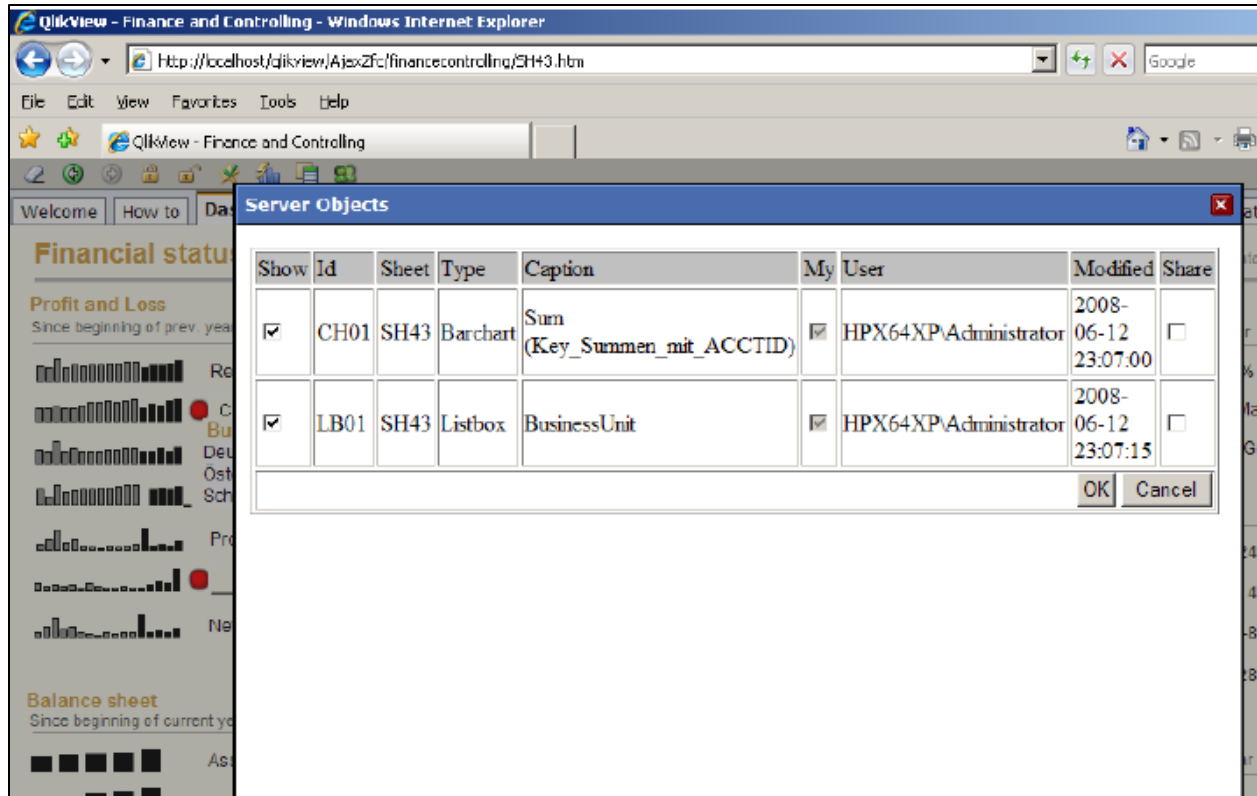
EX1009, 0009, 0013, 0065, 0114-15, 0140, 0158.

390. Shown below, a user accessing QlikView via a QlikView plug-in can interface with the Server Objects dialog to share selected objects with other users. EX1009, 0140.



EX1009, 0140.

391. Similarly, the AJAX client can also access and share server objects with other users. EX1009, 0158.



EX1009, 0158.

392. Indeed, “[o]ne of the major design goals for QlikView 8 is to enhance user collaboration during the QlikView experience.” EX1009, 0013.

393. For at least these reasons, the QlikView Manuals disclose the additional limitation in claims 6, 12, and 18.

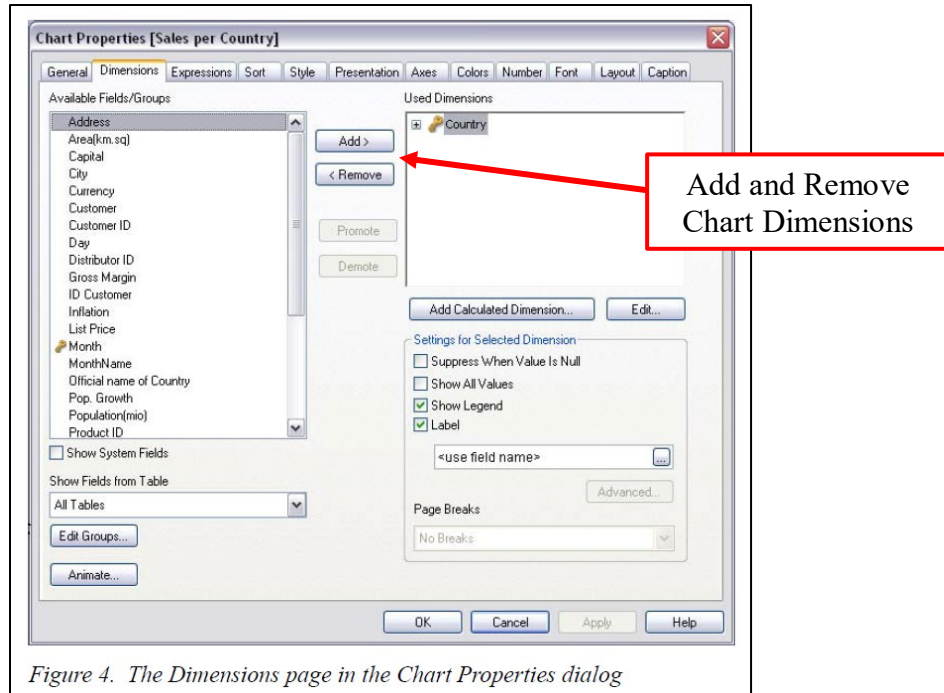
XIV. Ground 10: The QlikView Manuals in view of Couckuyt renders obvious claims 3, 9, and 15.

394. The QlikView Manuals in view of Couckuyt teach or suggest every element in claims 3, 9, and 15.

A. A POSA would have been motivated to combine the teachings in the QlikView Manuals and Couckuyt.

395. As I explained in Section XIII.A, a POSA would have been motivated to combine the QlikView Manuals. A POSA would have also been motivated to combine the QlikView Manuals and Couckuyt.

396. As Ground 9 shows, the QlikView Manuals describe a system that generates an interactive chart using a Chart Wizard with different dialog boxes (e.g., one dialog box for selecting chart data and a second dialog box for selecting a chart theme). In the dialog box below, a user must add available fields/groups to the chart's dimensions. As depicted, the user interacts with the Add and Remove buttons to include or exclude fields/groups from the chart's dimensions. EX1008, 0897. However, deciding which field(s) to include and the order was not always intuitive. Also, it was not always clear whether adding a dimension would do anything because extraneous dimensions are discarded. EX1008, 0895.



EX1008, 0894 (annotated).

397. Because the QlikView interface is not as intuitive as other design options, a POSA would have looked for other references about adding data to charts. This would have led to Couckuyt, which describes a data visualization tool that generates a chart by dragging-and-dropping data fields into “drop zones” of the chart. EX1006, 7:10-31, FIG. 4.

398. A possible combination is below, which is more intuitive than QlikView’s Add, Remove, Promote, and Demote options. Providing drag-and-drop functionality from a data pane (the available fields/groups) to a chart pane (top right) would have enabled the user to better know what data applies to which chart dimension. For example, to plot sales by country, the user could drag the country

data field to a drop zone (e.g., the x-axis) and the sales data field to a drop zone (e.g., the y-axis). This simple design change would have allowed users to view changes in real time.

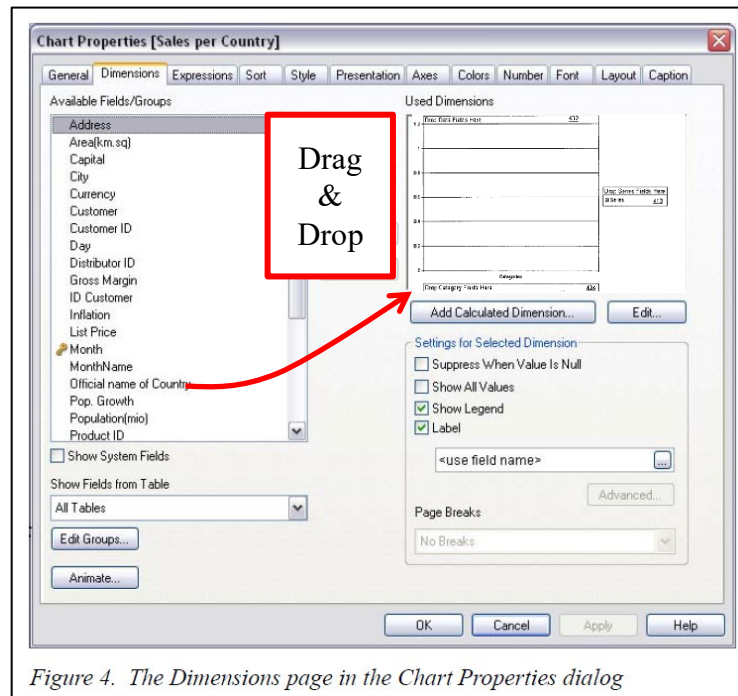


Figure 4. The Dimensions page in the Chart Properties dialog

EX1008, 0894 (modified in view of Couckuyt and annotated).

399. Combining QlikView's Chart Wizard and Couckuyt's drag-and-drop would have combined prior-art elements according to known methods to yield predictable results. Dividing a UI into multiple panes (e.g., one for data and one for a chart) was a basic building block of UI design. Indeed, most of QlikView's interfaces already used panes, including the one above. Drag-and-drop was also ubiquitous in the industry. Because panes and drag-and-drop were both well-

known, a POSA would have known how to predictably implement them within a data visualization tool.

400. Applying drag-and-drop to select data for a chart would have improved a similar device in the same way. Both QlikView and Couckuyt describe data visualization tools that create charts. *Compare* EX1008, 0887-89, *with* EX1006, 3:38-41. Indeed, both QlikView and Couckuyt describe creating the same types of charts such as bar charts and pie charts. *Compare* EX1008, 0887, *with* EX1006, 3:38-41. Both references source data from databases and spreadsheet documents using Microsoft products. *Compare* EX1008, 0061-63, 0242, 0249, *with* EX1006, 5:29-38. And both select data from a field list to populate charts. *Compare* EX1008, 0894-95, *with* EX1006, 1:29-31. Additionally, a POSA would have had a reasonable expectation of success because Couckuyt teaches that its system can work on “any program module operative to graphically render charts or graphs or any other graphical display of data.” EX1006, 5:27-29.

B. Claim 3

401. Claim 3 depends from claim 2 and recites “*wherein the user interface further includes a data pane that allows for the first user to drag the chart data from the data pane to the chart pane, wherein the interactive chart is generated after the chart data is dragged from the data pane to the chart pane.*” The QlikView Manuals in view of Couckuyt disclose this limitation.

402. The QlikView Manuals disclose “*wherein the user interface further includes a data pane*” as shown below. However, the QlikView Manuals do not explicitly disclose drag-and-drop of data from the data pane into a chart pane. Couckuyt provides this teaching.

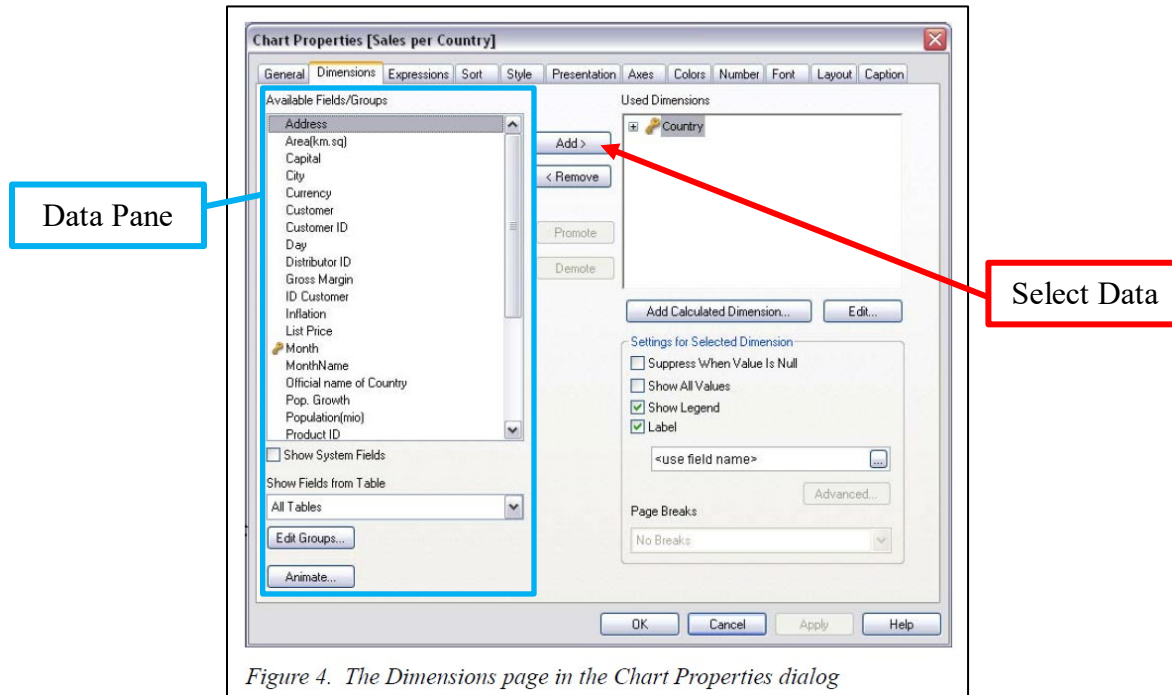
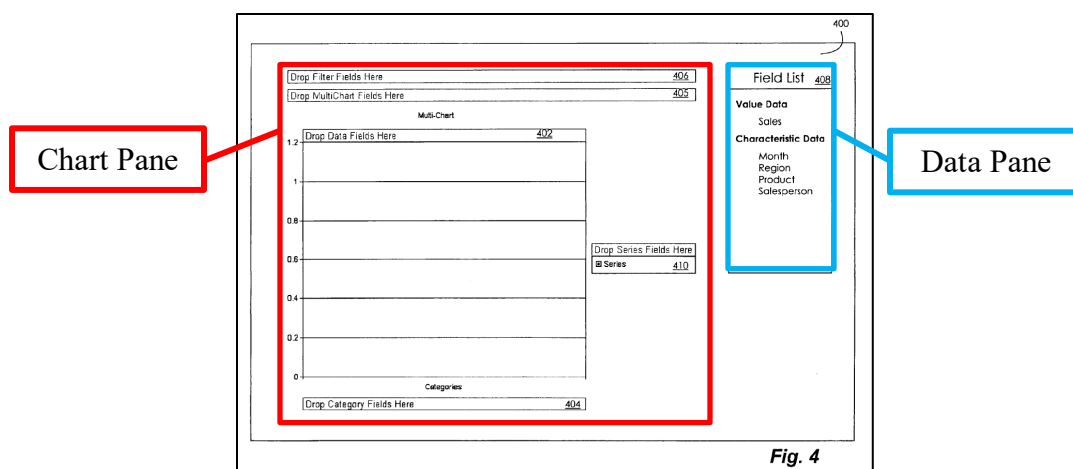


Figure 4. The Dimensions page in the Chart Properties dialog

EX1008, 0894 (annotated).

403. Shown below, Couckuyt discloses a drag-and-drop UI (DDUI), which has “*a chart pane*” and “*a data pane.*”



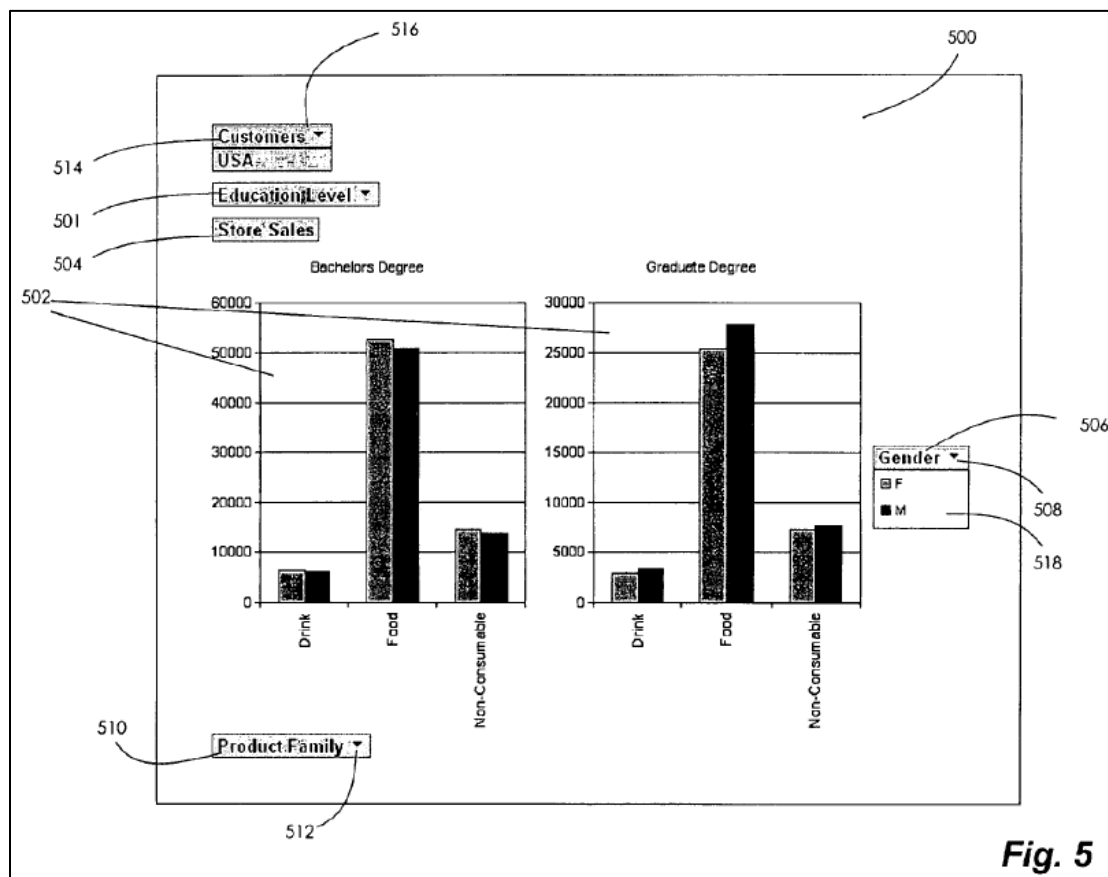
EX1006, FIG. 4 (annotated).

404. Couckuyt’s DDUI “allows for the first user to drag the chart data from the data pane to the chart pane.” It “permits the value data fields and the characteristic data fields (collectively, data fields) to be dragged and dropped into drop zones.” EX1006, 7:24-26; *see also* EX1006, 1:14-34, 2:22-32, 2:42-55, 7:64-8:36, FIGs. 4-6.

405. Couckuyt’s DDUI also discloses “wherein the interactive chart is generated after the chart data is dragged from the data pane to the chart pane.” Couckuyt (like the QlikView Manuals) stores data from databases and spreadsheet documents into a data module. EX1006, 4:16-37, 5:25-38, FIGs. 1, 2. When the DDUI receives an indication that the user has dropped data into the chart pane (e.g., the user releases the click of mouse 42), the DDUI automatically generates the interactive chart from the stored data. Couckuyt explains: “Dropping a field icon into a drop zone causes the charting DDUI to populate a corresponding

portion of the chart with the data from the field.” EX1006, 1:32-34; *see also* EX1006, 7:26-39, 7:64-8:36, 9:48-52, 10:52-54, FIGs. 5, 6. Indeed, Couckuyt states it will “automatically generate” the charts. EX1006, 2:12-16, 3:23-28, 7:14-16, Abstract.

406. For example, Couckuyt discloses generating multiple charts for each sub-data field included in dropped data. EX1006, 8:2-4. As shown below, when “Education Level” data is dropped, sub-data fields such as “Bachelors Degree” and “Graduate Degree” are plotted in separate charts. EX1006, 8:5-10.



EX1006, FIG. 5.

407. A POSA would have been motivated to combine the QlikView Manuals and Couckuyt to perform this drag-and-drop function for the reasons in Section XIV.A. The QlikView Manuals in view of Couckuyt therefore disclose the additional limitation in claim 3.

C. Claim 9

408. Claim 9 depends from claim 8 and recites “*receiving an indication that the chart data has been dragged from a data pane in the user interface and dropped into the chart pane; and generating the interactive chart after the chart data is dropped into the chart pane.*” The QlikView Manuals in view of Couckuyt disclose these limitations for the reasons I explained for claim 3.

409. A POSA would have been motivated to combine the QlikView Manuals and Couckuyt to perform this drag-and-drop function for the reasons in Section XIV.A. The QlikView Manuals in view of Couckuyt therefore disclose the additional limitation in claim 9.

D. Claim 15

410. Claim 15 depends from claim 14 and recites “*wherein the program is further executable to: allow the first user to drag the chart data from a data pane of the user interface and drop the data into the chart pane; and generate the interactive chart after the chart data is dropped into the chart pane.*” The QlikView Manuals in view of Couckuyt disclose these limitations for the reasons I

explained for claim 3. The drag-and-drop operation is performed by “*the first user*” because it is performed on “*the first website*.”

411. A POSA would have been motivated to combine the QlikView Manuals and Couckuyt to perform this drag-and-drop function for the reasons in Section XIV.A. The QlikView Manuals in view of Couckuyt therefore disclose the additional limitation in claim 15.

XV. No Objective Indicia of Nonobviousness

412. 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.

413. 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.

414. Considering the information I have reviewed, I am not aware of any objective indicia that might support the patentability of the claims of the ’595 patent. Further, from my review of the file history (EX1002) of the ’595 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.

415. Therefore, it is my opinion that no objective indicia of nonobviousness support the patentability of the claims of the '595 patent.

XVI. 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' followed by a stylized surname.

Christopher Schmandt

Date: 26 September 2024