(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 March 2007 (22.03.2007)

PCT

(10) International Publication Number
WO 2007/032925 A1

(51) International Patent Classification:
G06F 17/00 (2006.01) *G06F 3/14* (2006.01)
G06F 17/21 (2006.01)

(21) International Application Number:
PCT/US2006/034121

(22) International Filing Date: 29 August 2006 (29.08.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/716,293 12 September 2005 (12.09.2005) US
11/318,305 23 December 2005 (23.12.2005) US

(71) Applicant *(for all designated States except US)*: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(72) Inventors: **KOTHARI, Nikhil**; One Microsoft Way, Redmond, Washington 98052-6399 (US). **LE ROY, Bertrand**; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

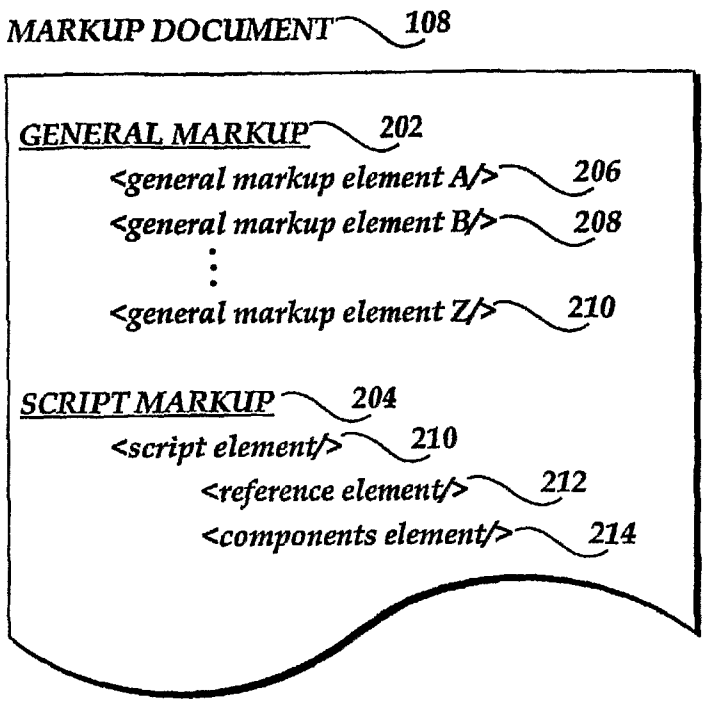
(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:
— *with international search report*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SCRIPT MARKUP



(57) Abstract: A script markup language provides a declarative mechanism for defining script based interactive behavior and application logic associated with a document. The script markup defining the interactive behavior and application logic is presented as an independent portion of the markup for the document, separated from any markup concerning the content and presentation of the document.

WO 2007/032925 A1

SCRIPT MARKUP

BACKGROUND

Historically, markup was used to refer to the process of marking manuscript copy for typesetting with directions for formatting such as use of type fonts and sizes, spacing, indentation, etc. In today's digital age, markup refers to electronic markup, i.e., the internal and sometimes invisible codes in an electronic document that describe the formatting of the document. Generally, a user can view the markup of an electronic document by looking at the source code of the document with the browser displaying the electronic document. The electronic markup of a document generally provides encoding of text as well as details about the structure, appearance and presentation of the text and content in the document.

The markup of an electronic document usually is programmed using a markup language. A markup language provides syntax and procedures for embedding in a document tags that control the formatting of the text when the document is viewed by a special application such as a Web browser. Commonly used electronic markup languages include HTML, XML, and ASP.NET. Traditionally, markup languages are used to design the content and appearance of a static document.

However, for an interactive application such as a Web application, the content and/or presentation of a document such as a Web page may change, for example, based on user input. The markup of the document thus needs to be accompanied by information governing the behavior of the document. Traditionally, document behavior has been implemented procedurally in a script. To provide dynamic document behavior, a markup of the document may call on methods in the script at the appropriate time. The intermingling of markup and calls to script methods thus makes it difficult to independently design the markup for a document. Meanwhile, because a script language traditionally has been procedural and imperative, a user of a document usually cannot use the script language to design a specific behavior for the document.

While specific disadvantages of existing systems have been illustrated and described in this Background Section, those skilled in the art and others will recognize that the subject matter claimed herein is not limited to any specific implementation for solving any or all of the described disadvantages.

SUMMARY

This summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This summary is not intended to identify key features of the claimed subject matter, nor is it intended to be used as an aid in determining the scope of the claimed subject matter.

Aspects of the invention make available a script markup language that provides a declarative mechanism for defining script-based interactive behavior and application logic associated with a document. Aspects of the invention also enable the interactive behavior and application logic associated with a document to be defined as an independent layer of the document ("script markup"), separated from markup concerning the content and presentation of the document ("general markup").

One aspect of the invention employs a script markup language to program script markup for a document to define the behavior of the document. The script markup may be included or referenced in a markup document containing markup information for displaying a document. The markup document may further include a general markup portion including one or more general markup elements defining the content and/or the appearance of the document to be displayed. The general markup portion and the script markup portion are separated from each other in the markup document, though the script markup portion may define behaviors of the general markup elements in the general markup portion.

In accordance with another aspect of the invention, the script markup portion includes one or more script markup elements. For example, the script markup elements may include a script element that contains a reference element and a components element. The reference element may include one or more references to script files used by the script markup portion. The components element may define one or more script objects for controlling the behavior of the document to be displayed.

In accordance with yet another aspect of the invention, a script object may contain one or more attributes such as a property attribute, a method attribute, an event attribute, or a reference to another element in the markup document. For example, a script object

may reference a general markup element in the general markup portion so the script object can control the behavior of the general markup element. A script object may also reference another script object defined or referenced by the script markup portion. A script object may also contain one or more sub-script objects such as an event object, a binding object, and an action object. An event handler may be provided for an event object. The event handler may connect the script markup with developer-defined code.

In accordance with a further aspect of the invention, a script object may communicate with another script object. For example, a binding object associated with a script object may bind a property of the script object with the property of another script object. In addition, an action object associated with a script object may perform a specific action upon the occurrence of a specific event. The specific action may be to execute a method associated with another script object or to configure a property associated with another script object.

DESCRIPTION OF THE DRAWINGS

The foregoing aspects and many of the attendant advantages of this invention will become more readily appreciated as the same become better understood by reference to the following detailed description, when taken in conjunction with the accompanying drawings, wherein:

FIGURE 1 is a block diagram illustrating an exemplary computing system for implementing aspects of the invention;

FIGURE 2 is a block diagram illustrating an exemplary partition of a markup document according to one aspect of the invention; and

FIGURE 3 is a text diagram illustrating an exemplary markup document implementing aspects of the invention.

DETAILED DESCRIPTION

The following text illustrates and describes exemplary embodiments of the invention. However, those of ordinary skill in the art will appreciate that various changes can be made therein without departing from the spirit and scope of the invention.

FIGURE 1 illustrates an exemplary computing system 100 for implementing aspects of the invention. The computing system 100 includes a server component 102 and a client component 104. Generally, a browser 106 is associated with the client 104 for displaying a document such as a Web page. In a typical scenario, when the browser 106

requests to display a document, e.g., a Web page, the client 104 sends a document request to the server 102. The server 102 then sends the client 104 the markup document 108 containing markup information for displaying the requested document. The markup document 108 may exist in a database 110 associated with the server 102. Often, the server 102 and the client 104 exist on the same computer system. Alternatively, they may exist on different computer systems and communicate through a network (not shown).

In embodiments of the invention, upon receiving the markup document 108, the browser 106 parses and interprets the markup document 108 to display the requested document according to the definitions provided in the markup document 108.

In exemplary embodiments of the invention, the markup document 108 for a document such as a Web page provides general markup that defines the content and/or presentation of the document. The markup document 108 further includes or references script markup that defines the behavior of the document. FIGURE 2 illustrates exemplary blocks of information presented in the markup document 108. As shown in FIGURE 2, the markup document 108 includes a general markup portion 202 and a script markup portion 204.

The general markup portion 202 defines the formatting of the content and/or the overall appearance of the document to be displayed. The general markup portion 202 may define one or more general markup elements. For example, FIGURE 2 illustrates that the general markup portion 202 includes multiple general markup elements such as a general markup element A (206), a general markup element B (208), and a general markup element Z (210).

On the other hand, content of the script markup portion 204 defines interactive behavior and application logic associated with the document to be displayed. In embodiments of the invention, the content of the script markup portion 204 defines or references one or more script objects, and instantiates the script objects along with attributes defining the states, property values of the script objects. As shown in FIGURE 2, in embodiments of the invention, the script markup portion 204 is separated from the general markup portion 202 and is an independent portion of the markup document 108. Alternatively, in some embodiments of the invention, the script markup portion 204 can be included in a separate file, which is then referenced by the markup document 108. As shown in FIGURE 2, the content of the script markup portion 204 includes multiple script markup elements such as a script element 210, a reference

element 212, and a components element 214. Both the general markup elements and the script markup elements are called markup elements.

In an exemplary embodiment of the invention, the script element 210 defines the overall scope of the script markup portion 204. All other elements in the script markup portion 204, such as the reference element 212 and the components element 214, are contained within the script element 210. Referring back to FIGURE 1, while interpreting the script markup portion 204, the browser 106 navigates through the script element 210 to interpret the included definitions, so to decide the behavior of the document to be displayed.

In embodiments of the invention, the reference element 212 references script files external to the markup documents 108 that are used by markup elements in the markup documents 108. The external script files may detail dependency information that the markup elements may use. Preferably, the external script files may also provide implementation details of script markup elements defined or referenced in the script markup portion 204.

The components element 214 contains one or more script object definitions that actually define the behavior of the document to be displayed. In exemplary embodiments of the invention, one or more of the script objects defined in the components element 214 may reference and hence define behaviors of one or more of the general markup elements included in the general markup portion 202.

FIGURE 3 illustrates an exemplary markup document 108 implementing the exemplary markup elements illustrated in FIGURE 2. As shown in FIGURE 3, the exemplary markup document 108 contains a hierarchical structure, in which one markup element may be contained by another markup element. Each markup element includes tags, as denoted by, for example, <> symbols, with the actual element being detailed between the tags. Each markup element includes a start tag and an end tag, wherein a start tag begins a markup element and an end tag ends the corresponding markup element. For example, as shown in FIGURE 3, the script element 210 begins with the start tag <> on line 3 and ends with the end tag </> on line 34. As will be described in detail below, the markup elements in the markup document 108 further contain one or more attributes with assigned values.

The exemplary markup document 108 shown in FIGURE 3 illustrates script-defined behavior of two counters. As shown in FIGURE 3, lines 1-2 illustrate an

exemplary general markup portion 202. Here, two general markup elements—Counter#1 and Counter#2—are defined, wherein Counter#1 has an "id" attribute with the value "counterLabel1" and Counter#2 has an "id" attribute with the value "counterLabel2."

Lines 3-34 illustrate an exemplary script markup portion 204 that specifies the behavior of the two counters defined in lines 1-2. Specifically, line 3 signals the beginning of a definition for an exemplary script element 210 and line 34 signals the end of the definition. The exemplary script element 210 includes an exemplary reference element 212 (lines 5-8) that links in two JavaScript files—AtlasUI.js and AtlasControls.js. Lines 9-32 illustrate an exemplary components element 214 that defines a plurality of script objects. For example, line 10 defines a script object Counter 302 that is identified as "counter1," while line 11 defines a script object Counter 304 that is identified as "counter2" and has a value of "10000." The code between lines 12-16 and lines 17-21 each defines a script object Timer (306, 316) that periodically, e.g., every 500 seconds, enables an event object Tick (308, 318). In embodiments of the invention, a script object may include one or more sub-script objects. For example, the script object Timer 306 includes an event object Tick 308, which further includes an action object invokeMethod 310. For another example, the script object Label 312 defined in lines 22-26 includes a binding object 314.

In exemplary embodiment of the invention, a script object may be associated with one or more attributes whose values are used to define the behavior of the script object. An attribute can be, for example, a property, a method, or an event associated with the script object. An attribute may also be a reference to another markup element. For example, the script object Counter 304 defined in line 11 has a property attribute "id" and a property attribute "value". The action object invokeMethod 310 defined in line 14 has an method attribute "Method" that is set to an exemplary "increment" method. For example, instead of using an event object Tick 308, the script object Timer 306 may have an event attribute "Tick". The scrip object Label 312 defined in line 22 has an attribute "targetElement" that references the general markup element Counter#1 identified as "counterLabel1" in line 1.

In exemplary embodiments of the invention, a script object may reference a general markup element defined in the general markup portion 202 of the markup document 108 and define document behavior associated with the referenced general markup element. For example, the code between lines 22-26 defines a script object

Label 312 that references the general markup element Counter#1 defined in line 1. The code between lines 27-31 defines a script object Label 320 that references the general markup element Counter#2 defined in line 2. Consequently, the script objects Label 312 and Label 320 may specify the behaviors of the general markup elements Counter#1 and Counter#2 in the general markup portion 202.

In embodiments of the invention, a script object may communicate with another script object by performing a specific action upon occurrence of a specific event. For example, in embodiments of the invention, a script object may be associated with an event, the occurrence of which initiates a corresponding event handler, which may link to developer-defined code for markup elements in the markup document 108. In an exemplary embodiment of the invention, the event handler includes one or more specific actions to be performed on one of the script objects in the components element 214. An exemplary action can be to invoke a method associated with another script object. Another exemplary action can be to configure a property associated with another script object. In a typical embodiment of the invention, both the event and the action are also script objects including one or more attributes. For example, the script object Timer 306 contains an event object Tick 308, the enablement of which initiates an action object invokeMethod 310. The action object invokeMethod 310 has an attribute "target" specifying a target script object—"counter1", for example—and an attribute "method" specifying the function to be performed on the target script object.

Another exemplary mechanism for one script object to communicate with another script object is a binding mechanism that connects a property of one script object with a property of another script object; the change of one property thus is reflected on the other property. For example, as shown in FIGURE 3, the script object Label 312 includes a binding object 314. The binding object 314 has an attribute "dataContext" that specifies the script object and an attribute "dataPath" that specifies one of the script object's properties with which the script object Label 312 will bind its property "text". As a result of the binding, the value of the script object Counter 302 defined in line 10 is reflected in the "text" property associated with the script object Label 312 and hence is displayed in the general markup element Counter#1 defined in line 1. In an exemplary embodiment of the invention, a binding object provides a transform functionality that transforms the type of the property that provides the data into the type of the property that receives the data.

For example, the transform functionality for the binding object 314 may convert the type of the property specified by "dataPath" into the type of the property specified by "text".

It is to be understood that FIGURE 3 illustrates only exemplary formats of a script markup language for implementing aspects of the invention. These exemplary formats should be used for illustration purposes only. These exemplary formats do not limit the script markup language offered by embodiments of the invention to the specific formats, syntax, and functionalities illustrated. For example, the exemplary markup document 108 has been illustrated using XML syntax and formats. However, those of ordinary skill in the art will appreciate that aspects of the invention may be implemented in different markup languages such as HTML, ASP.NET, JavaScript Object Notation, etc.

In embodiments of the invention, a developer may custom define a script object model. The script object model, for example, specifies attributes, such as property, method, and/or event attributes, and any sub-script object models that may be associated with the script object model. The script object model then is registered with the browser 106, for example, through a type manager associated with the browser 106. The browser 106 thus knows how to interpret and process a script object instantiated based on the script object model. As a result, the script markup language provided by aspects of the invention is extensible in that new script object models can be defined and registered with a browser for interpreting script markups containing script objects instantiated based on the script object models.

Although aspects of the invention have been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the specific features or acts described above. Rather, the specific features and acts described above are disclosed as example forms of implementing the claims.

CLAIMS

The embodiments of the invention in which an exclusive property or privilege is claimed are defined as follows:

1. A markup document (108) stored on a computer-readable medium, comprising:

a general markup portion (202) that includes one or more general markup elements; and

a script markup portion (204) that is separated from the general markup portion (202) and content of the script markup portion (204) includes one or more script markup elements;

wherein content of the script markup portion (204) defines behavior of a document and the general markup portion (202) defines content and/or appearance of the document.

2. The markup document of Claim 1, wherein one of the one or more script markup elements is a reference element containing one or more references to existing script files.

3. The markup document of Claim 1, wherein one of the one or more script markup elements is a components element that defines one or more script objects.

4. The markup document of Claim 3, wherein one of the one or more script objects references one of the one or more general markup elements in the general markup portion.

5. The markup document of Claim 4, wherein one of the one or more script objects includes one or more attributes.

6. The markup document of Claim 5, wherein one of the one or more attributes is one of a property attribute, a method attribute, and an event attribute.

7. The markup document of Claim 3, wherein one of the one or more script objects includes one or more script objects ("sub-script objects").

8. The markup document of Claim 7, wherein one of the sub-script objects is a binding object for connecting an attribute of the script object with an attribute of another script object, wherein both attributes are property attributes.

9. The markup document of Claim 8, wherein the binding object includes a function for converting type of the attribute of the script object into type of the attribute of the another script object.

10. The markup document of Claim 7, wherein one of the sub-script objects is an event object.

11. The markup document of Claim 10, wherein the event object further includes an event handler detailing what to do when the event occurs.

12. The markup document of Claim 11, wherein the event handler includes an action object that initiates a specific action when the event occurs.

13. The markup document of Claim 12, wherein the action involves executing an attribute of another script object, wherein the attribute is a method attribute.

14. The markup document of Claim 12, wherein the action involves configuring an attribute of another script object, wherein the attribute is a property attribute.

15. The markup document of Claim 1, wherein the content of the script markup portion is stored in a file and wherein the script markup portion in the markup document is a reference to the file.

16. A computing system (100) including a browser (106), a computer-implemented method for interpreting a markup document (108) to display a document is provided, the method comprising:

receiving the markup document (108), wherein the markup document (108) contains a script element (210); and

interpreting the markup document (108) by:

regarding content contained within the script element (210) as script markup (204) defining behavior of the document to be displayed.

17. The method of Claim 16, wherein the script element includes a reference element, the method further comprising: retrieving content of a file referenced within the reference element.

18. The method of Claim 16, wherein the script element includes a components element, the method further comprising: interpreting content within the components element as one or more script objects defining the behavior of the document to be displayed.

19. The method of Claim 18, wherein the markup document further includes a general markup portion containing one or more general markup elements for defining content and/or appearance of the document, at least one of the one or more script objects represents at least one of the one or more general markup elements and defines behavior of the general markup element.

20. The method of Claim 18, wherein the browser interprets the one or more script objects by using one or more custom-defined script object models.

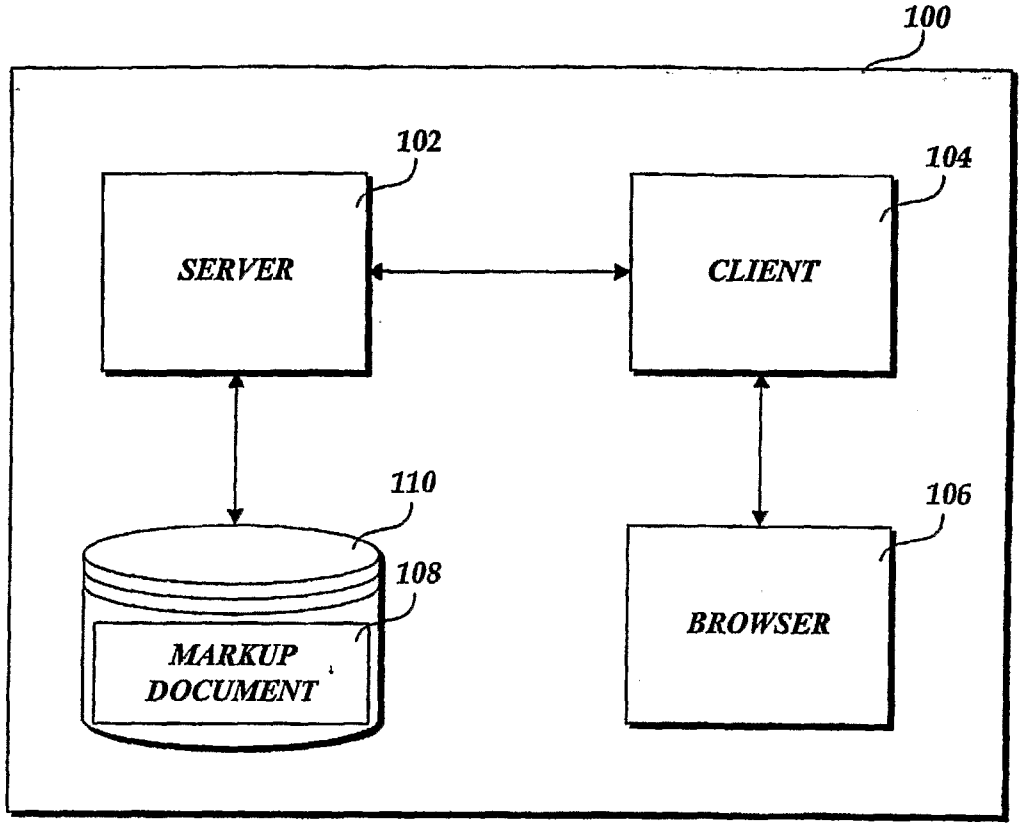

100
102
SERVER
104
CLIENT
110
108
MARKUP DOCUMENT
106
BROWSER

Fig.1.

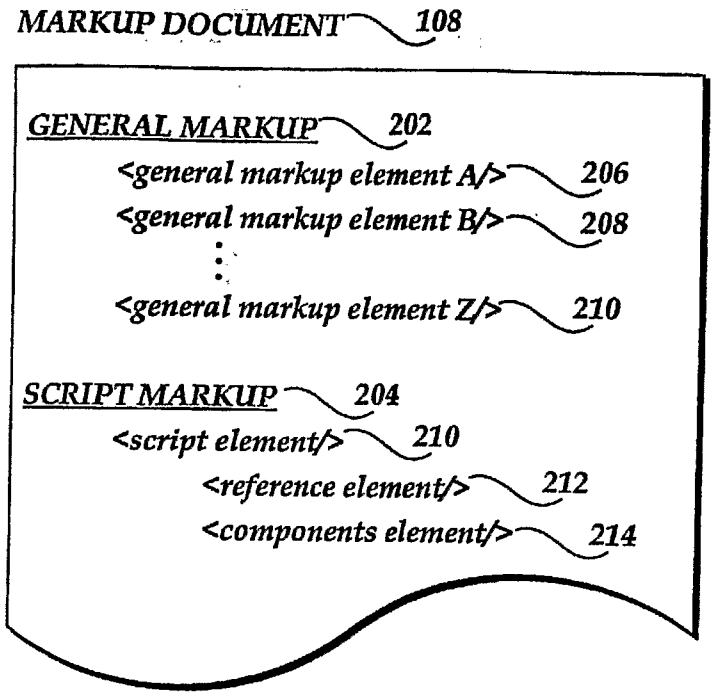

MARKUP DOCUMENT 108
GENERAL MARKUP 202
<general markup element A/> 206
<general markup element B/> 208
<general markup element Z/> 210
SCRIPT MARKUP 204
<script element/> 210
<reference element/> 212
<components element/> 214

Fig.2.

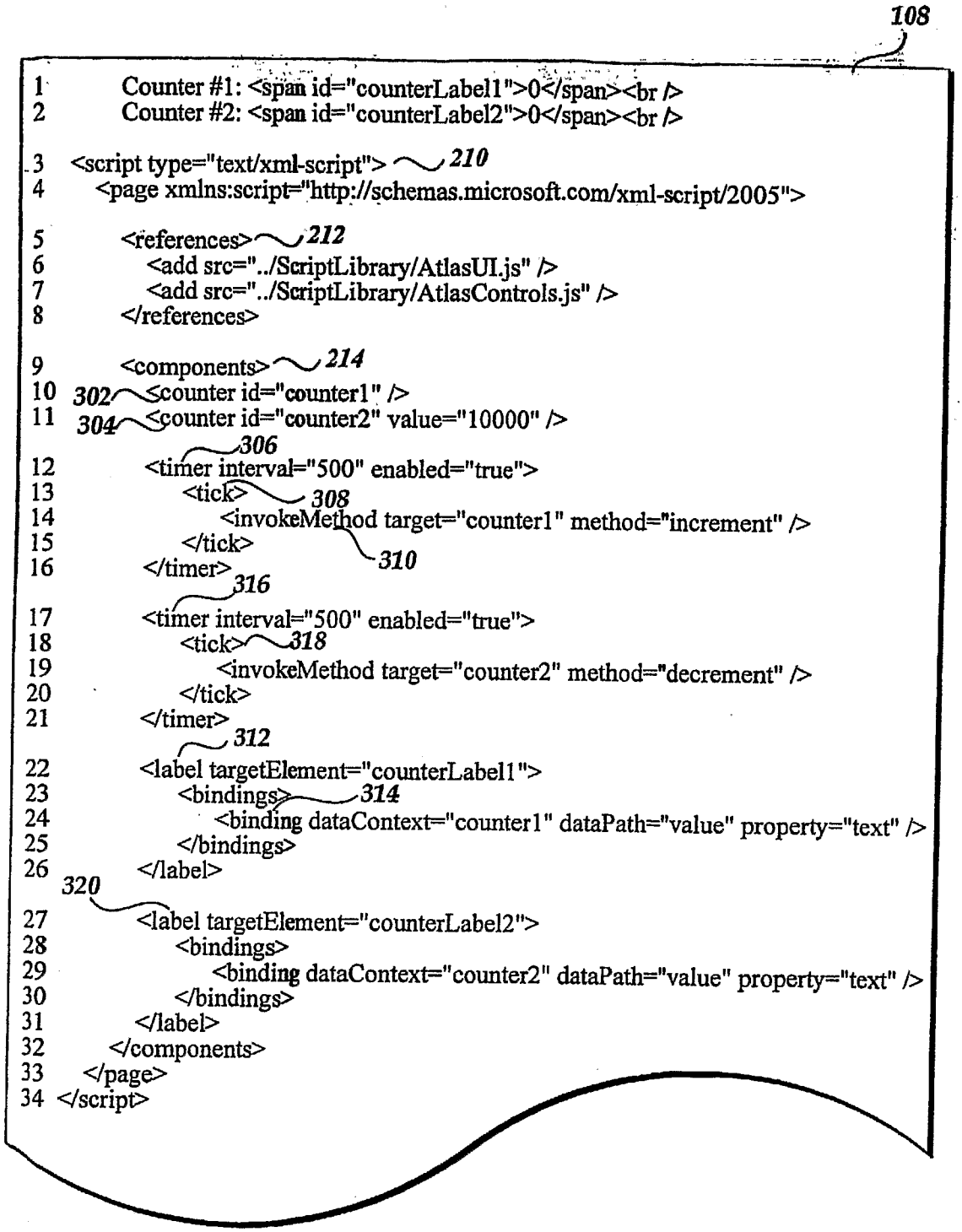

108
1 Counter #1: 0
2 Counter #2: 0
3 <script type="text/xml-script"> 210
4 <page xmlns:script="http://schemas.microsoft.com/xml-script/2005">
5 <references> 212
6 <add src="../ScriptLibrary/AtlasUI.js" />
7 <add src="../ScriptLibrary/AtlasControls.js" />
8 </references>
9 <components> 214
10 302 <counter id="counter1" />
11 304 <counter id="counter2" value="10000" />
306
12 <timer interval="500" enabled="true">
13 <tick> 308
14 <invokeMethod target="counter1" method="increment" />
15 </tick> 310
16 </timer>
316
17 <timer interval="500" enabled="true">
18 <tick> 318
19 <invokeMethod target="counter2" method="decrement" />
20 </tick>
21 </timer>
312
22 <label targetElement="counterLabel1">
23 <bindings> 314
24 <binding dataContext="counter1" dataPath="value" property="text" />
25 </bindings>
26 </label>
320
27 <label targetElement="counterLabel2">
28 <bindings>
29 <binding dataContext="counter2" dataPath="value" property="text" />
30 </bindings>
31 </label>
32 </components>
33 </page>
34 </script>

Fig.3.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2006/034121

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/00(2006.01)i, G06F 17/21(2006.01)i, G06F 3/14(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC8 G06F 17/00, G06F 17/21, G06F 3/14

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean patents and applications for inventions since 1975.
Korean utility models and applications for utility models since 1975.
Japanese utility models and applications for utility models since 1975.

Electronic data base consulted during the intertnational search (name of data base and, where practicable, search terms used)

e-KIPASS(KIPO internal) "markup document, script markup, content"

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 0190861 A1 (CHUNG HYUN-KWON, PARK SUNG-WOOK) 30 SEPTEMBER 2004 See abstract; figures 3,4; claims 1 - 23.	1 - 20
A	WO 03/036460 A1 (SAMSUNG ELECTRONICS CO LTD) 1 MAY 2003 See abstract; figures 1,2; claims 1 - 7.	1 - 20
A	WO 03/100548 A2 (KONINKLIJKE PHILIPS ELECTRONICS N.V. et al.) 04 DECEMBER 2003 See abstract; figures 1,2; claims 1 - 10.	1 - 20

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents,such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 JANUARY 2007 (11.01.2007)	**11 JANUARY 2007 (11.01.2007)**

Name and mailing address of the ISA/KR

Korean Intellectual Property Office
920 Dunsan-dong, Seo-gu, Daejeon 302-701,
Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

KIM, MYOUNG CHAN

Telephone No. 82-42-481-5783

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.
PCT/US2006/034121

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US2004190861A1	30.09.2004	BR0406218A	09.08.2005
		CA2498692A1	14.10.2004
		US2004190861A1	30.09.2004
		US2004190875A1	30.09.2004
		US2004190876A1	30.09.2004
		WO2004088662A1	14.10.2004
WO2003036460A1	01.05.2003	BR200213454A	13.10.2004
		CA2465104AA	01.05.2003
		CA2465104A1	01.05.2003
		CN1275134C	13.09.2006
		CN1599897A	23.03.2005
		CZ20040509A3	18.08.2004
		EP1454226A1	08.09.2004
		EP1454226A4	29.12.2004
		EP1491994A1	29.12.2004
		JP17506785	03.03.2005
		JP2005506785T2	03.03.2005
		KR1020030034034	01.05.2003
		KR2003034034A	. .A
		MXPA04003815A	30.07.2004
		PL369579A1	02.05.2005
		RU2004112416A	10.04.2005
		US2003194207A1	16.10.2003
		US2003194207AA	16.10.2003
		US2006197493AA	07.09.2006
WO2003100548A2	04.12.2003	AU2003228026A1	12.12.2003
		AU2003228026A8	12.12.2003
		AU2003228026AA	12.12.2003
		AU2003228026AH	12.12.2003
		CN1656481A	17.08.2005
		EP1552357A2	13.07.2005
		GB200211897A0	03.07.2002
		JP17527029	08.09.2005
		JP2005527029T2	08.09.2005
		KR1020040111641	31.12.2004
		US20050204280A1	15.09.2005
		US2005204280AA	15.09.2005
		WO2003100548A3	21.05.2004