

"	"	1	2	3	4	5	6	7	8	9
"a	a"	q	a	d	g	j	m	r	t	w
"a	b"	q	a	e	i	l	o	r	u	w
"a	c"	q	a	e	h	k	o	r	t	w
"a	d"	q	a	e	i	j	m	r	v	y
"a	e"	q	a	d	g	j	m	p	t	w
"a	f"	q	a	f	g	j	m	r	t	w
"a	g"	q	a	e	i	j	o	r	t	w
"a	h"	q	a	e	g	j	m	p	u	w
"a	i"	q	a	d	g	l	n	r	t	w
"a	j"	q	a	d	g	j	o	p	t	w
"a	k"	q	a	e	i	j	m	s	t	w
"a	l"	q	a	e	i	l	o	s	t	w
"a	m"	q	a	e	i	l	o	p	u	w
"a	n"	q	a	d	g	k	n	s	t	y
"a	o"	q	a	d	h	j	m	p	t	w
"a	p"	q	a	e	g	l	o	p	t	w
"a	q"	q	a	d	g	j	o	s	t	w
"a	r"	q	a	e	g	k	o	r	t	y
"a	s"	q	a	e	i	k	o	s	t	y
"a	t"	q	a	e	i	l	o	s	t	w
"a	u"	q	a	d	g	l	m	s	t	w
"a	v"	q	a	e	i	j	o	p	t	w
"a	w"	q	a	d	i	k	m	p	t	w
"a	x"	q	a	e	i	l	m	s	t	w
"a	y"	q	b	e	i	j	m	s	t	w
"a	z"	q	a	d	i	j	m	p	t	w
"a	"	q	a	d	g	j	m	p	t	w
"b	a"	q	a	d	g	k	n	s	t	y
"b	b"	q	a	d	g	j	m	s	t	y
"b	c"	q	a	d	g	j	o	r	t	w
"b	d"	q	a	d	g	l	m	s	t	w
"b	e"	q	a	e	g	j	n	s	t	x
"b	f"	q	a	d	g	j	m	p	t	w
"b	g"	q	a	d	g	j	m	p	t	w
"b	h"	q	a	d	g	j	m	p	t	w
"b	i"	q	a	d	g	l	n	s	t	w
"b	j"	q	a	e	g	j	m	p	t	w

FIG. 10A

"	.	1	2	3	4	5	6	7	8	9
"b	k"	q	a	d	g	j	m	p	t	w
"b	l"	q	a	e	i	j	o	p	t	y
"b	m"	q	a	d	g	j	m	r	t	w
"b	n"	q	a	d	g	j	m	p	t	w
"b	o"	q	a	d	i	j	o	s	u	x
"b	p"	q	a	d	g	j	m	s	t	w
"b	q"	q	a	d	g	j	m	p	t	w
"b	r"	q	a	e	i	j	o	p	u	w
"b	s"	q	c	d	i	j	o	s	t	w
"b	t"	q	a	d	g	j	m	p	t	w
"b	u"	q	a	d	i	l	m	s	t	y
"b	v"	q	a	d	i	j	m	p	t	w
"b	w"	q	a	e	g	j	m	p	t	w
"b	x"	q	a	d	g	j	m	p	t	w
"b	y"	q	a	d	g	j	m	p	t	w
"b	z"	q	a	d	g	j	m	p	t	w
"b	"	q	a	d	g	j	m	p	t	w
"c	a"	q	b	d	g	l	n	r	t	w
"c	b"	q	a	e	g	j	m	p	t	w
"c	c"	q	a	e	i	j	o	s	t	w
"c	d"	q	a	d	g	j	m	s	t	w
"c	e"	q	a	d	i	l	n	s	t	w
"c	f"	q	a	d	i	j	m	p	t	w
"c	g"	q	a	d	i	j	m	p	t	w
"c	h"	q	a	e	i	j	o	r	t	w
"c	i"	q	a	f	i	l	n	p	t	w
"c	j"	q	a	d	g	j	m	p	t	w
"c	k"	q	b	e	i	l	m	s	t	y
"c	l"	q	a	e	i	j	o	r	u	w
"c	m"	q	c	d	g	j	m	p	t	w
"c	n"	q	a	d	g	j	m	p	t	w
"c	o"	q	a	d	g	l	m	p	u	w
"c	p"	q	a	d	g	j	m	p	t	w
"c	q"	q	a	d	g	j	m	p	t	w
"c	r"	q	a	e	i	j	o	p	u	y
"c	s"	q	a	d	i	l	o	s	t	w
"c	t"	q	a	e	i	l	o	r	u	w

FIG. 10B

"-	-"	1	2	3	4	5	6	7	8	9
"c	u"	q	b	e	i	l	m	s	t	w
"c	v"	q	a	d	g	j	m	p	t	w
"c	w"	q	a	d	g	j	m	p	t	w
"c	x"	q	a	d	g	j	m	p	t	w
"c	y"	q	a	d	g	j	m	p	t	w
"c	z"	q	a	d	g	j	n	p	t	w
"c	"	q	a	d	g	j	m	p	t	w
"d	a"	q	b	d	g	j	n	r	t	y
"d	b"	q	a	d	g	j	m	p	t	w
"d	c"	q	a	d	g	j	m	p	t	w
"d	d"	q	a	e	i	l	m	r	t	w
"d	e"	q	a	d	g	l	m	r	v	x
"d	f"	q	a	d	i	j	m	p	t	w
"d	g"	q	a	e	g	j	m	p	t	w
"d	h"	q	a	e	g	j	m	p	t	w
"d	i"	z	a	f	g	l	n	s	t	w
"d	j"	q	a	d	g	j	m	p	t	w
"d	k"	q	a	d	g	j	m	p	t	w
"d	l"	q	a	e	i	j	m	p	t	y
"d	m"	q	a	e	i	j	m	s	t	w
"d	n"	q	a	e	i	j	m	p	t	w
"d	o"	q	a	e	i	j	n	p	t	w
"d	p"	q	a	d	g	j	m	p	t	w
"d	q"	q	a	d	g	j	m	p	u	w
"d	r"	q	a	e	i	j	o	p	t	w
"d	s"	q	a	e	i	j	m	p	t	w
"d	t"	q	a	e	g	j	m	r	t	w
"d	u"	q	a	e	g	l	m	r	t	w
"d	v"	q	a	d	i	j	m	p	t	w
"d	w"	q	a	d	i	j	o	p	t	w
"d	x"	q	a	d	g	j	m	p	t	w
"d	y"	q	a	d	g	j	m	p	t	w
"d	z"	q	a	d	g	j	m	p	t	w
"d	"	q	a	d	g	j	m	p	t	w
"e	a"	q	a	d	g	l	n	s	t	w
"e	b"	q	a	d	g	j	m	r	t	y
"e	c"	q	a	e	h	k	o	s	t	w

FIG. 10C

"	-	1	2	3	4	5	6	7	8	9
"e	d"	q	b	e	i	l	n	s	u	w
"e	e"	z	a	d	i	k	n	r	t	w
"e	f"	q	a	e	i	l	o	r	t	w
"e	g"	q	a	d	i	j	o	r	u	y
"e	h"	q	a	e	i	j	m	r	t	w
"e	i"	q	a	d	g	j	n	p	t	w
"e	j"	q	a	e	g	j	m	p	t	w
"e	k"	q	a	e	i	j	m	s	t	w
"e	l"	q	a	e	i	l	o	s	v	y
"e	m"	q	a	e	i	j	o	s	u	w
"e	n"	q	c	d	g	l	o	s	t	w
"e	o"	q	a	f	g	j	n	p	u	w
"e	p"	q	a	e	h	l	o	r	t	w
"e	q"	q	a	d	g	j	m	p	u	w
"e	r"	q	a	e	i	l	n	s	v	y
"e	s"	q	c	e	i	k	n	s	t	w
"e	t"	z	c	e	i	j	m	s	t	w
"e	u"	q	a	d	g	j	m	r	t	w
"e	v"	q	a	e	i	j	o	p	t	w
"e	w"	q	a	e	h	j	o	s	t	w
"e	x"	q	a	e	i	j	m	p	t	w
"e	y"	q	a	d	g	j	o	s	t	w
"e	z"	q	a	e	g	j	m	p	t	w
"e	"	q	a	d	g	j	m	p	t	w
"f	a"	q	c	d	i	k	m	s	v	x
"f	b"	q	a	d	g	j	m	p	t	w
"f	c"	q	c	d	g	j	m	p	t	w
"f	d"	q	a	d	g	l	m	p	t	w
"f	e"	q	a	e	i	j	o	r	t	w
"f	f"	q	a	d	g	j	m	p	t	w
"f	g"	q	a	d	g	j	m	p	t	w
"f	h"	q	a	d	g	l	m	p	t	w
"f	i"	q	c	e	g	l	n	r	t	x
"f	j"	q	a	d	g	j	m	p	t	w
"f	k"	q	a	d	g	j	m	p	t	w
"f	l"	q	a	e	i	j	o	p	t	y
"f	m"	q	a	d	g	j	m	p	t	w

FIG. 10D

"f	n"	1	2	3	4	5	6	7	8	9
"f	o"	q	a	d	g	j	m	p	t	w
"f	p"	q	c	d	g	l	n	r	u	w
"f	q"	q	a	d	g	j	m	p	t	w
"f	r"	q	a	d	g	j	m	p	t	w
"f	s"	q	a	e	g	j	o	p	t	y
"f	t"	q	a	d	g	j	m	p	t	w
"f	u"	q	a	d	g	l	n	r	t	w
"f	v"	q	a	d	g	j	m	p	t	w
"f	w"	q	a	d	g	j	m	p	t	w
"f	x"	q	a	d	g	j	m	p	t	w
"f	y"	q	a	d	g	j	m	p	t	w
"f	z"	q	a	d	g	j	m	p	t	w
"f	.	q	a	d	g	l	n	r	t	w
"g	a"	q	a	d	g	j	m	p	t	w
"g	b"	q	a	d	g	j	m	p	t	w
"g	c"	q	a	d	g	j	m	p	t	w
"g	d"	q	a	d	g	l	n	p	t	w
"g	e"	q	a	d	g	j	m	p	t	w
"g	f"	q	a	d	g	j	m	p	u	w
"g	g"	q	a	e	g	j	o	p	t	w
"g	h"	q	a	e	g	j	n	p	t	v
"g	i"	q	c	f	g	j	m	p	t	w
"g	j"	q	a	d	g	j	m	p	t	w
"g	k"	q	a	d	g	j	o	p	t	y
"g	l"	q	a	e	g	l	m	p	t	w
"g	m"	q	a	e	g	j	o	p	t	w
"g	n"	q	a	e	g	j	o	p	t	w
"g	o"	q	a	e	g	j	m	p	t	w
"g	p"	q	a	d	g	j	m	p	t	w
"g	q"	q	a	d	g	j	o	p	t	w
"g	r"	q	a	e	g	j	m	p	t	w
"g	s"	q	a	e	g	j	o	p	t	w
"g	t"	q	a	e	g	j	m	p	t	w
"g	u"	q	a	e	g	l	m	p	t	y
"g	v"	q	a	e	g	j	m	p	t	w
"g	w"	q	a	d	g	j	m	p	t	w

FIG. 10E

"	-	1	2	3	4	5	6	7	8	9
"g	x"	q	a	d	g	j	m	p	t	w
"g	y"	q	a	d	g	j	m	p	t	w
"g	z"	q	a	d	g	j	m	p	t	w
"g	"	q	a	d	g	j	m	p	t	w
"h	a"	q	b	d	g	j	n	p	t	w
"h	b"	q	a	d	g	j	m	p	t	w
"h	c"	q	a	d	g	j	o	p	t	w
"h	d"	q	a	d	g	j	m	p	t	w
"h	e"	q	c	d	g	j	n	r	t	y
"h	f"	q	a	d	g	j	m	p	t	w
"h	g"	q	a	d	g	j	m	p	t	w
"h	h"	q	a	d	g	j	m	p	t	w
"h	i"	q	c	e	g	j	n	s	t	w
"h	j"	q	a	d	g	j	m	p	t	w
"h	k"	q	a	d	g	j	m	p	t	w
"h	l"	q	a	d	g	j	m	p	t	y
"h	m"	q	a	e	g	j	m	p	t	w
"h	n"	q	a	d	g	j	o	p	t	w
"h	o"	q	b	d	g	j	n	r	u	w
"h	p"	q	a	d	g	j	m	p	u	w
"h	q"	q	a	d	g	j	m	p	t	w
"h	r"	q	a	e	g	j	o	p	t	w
"h	s"	q	a	d	g	j	m	p	t	w
"h	t"	q	a	d	g	l	m	s	t	w
"h	u"	q	a	d	g	l	m	r	t	w
"h	v"	q	a	d	g	j	m	p	t	w
"h	w"	q	a	d	g	j	m	p	t	w
"h	x"	q	a	d	g	j	m	p	t	w
"h	y"	q	a	d	g	j	m	p	t	w
"h	z"	q	a	d	g	j	m	p	t	w
"h	"	q	a	d	g	j	m	p	t	w
"i	a"	q	b	d	g	l	n	r	t	w
"i	b"	q	a	e	g	l	m	r	u	w
"i	c"	q	a	e	h	k	o	r	t	y
"i	d"	q	a	e	i	l	n	p	u	w
"i	e"	q	a	f	i	l	o	r	v	w
"i	f"	q	a			j				

FIG. 10F

"	"	1	2	3	4	5	6	7	8	9
"i"	g"	q	a	e	h	j	n	r	u	w
"i"	h"	q	a	d	g	j	m	p	t	w
"i"	i"	q	a	d	g	j	m	p	t	w
"i"	j"	q	a	d	g	j	m	p	t	w
"i"	k"	q	a	e	g	k	o	p	t	w
"i"	l"	q	a	e	i	l	o	s	t	y
"i"	m"	q	a	e	i	j	m	p	u	w
"i"	n"	q	c	e	g	k	n	s	t	w
"i"	o"	q	a	d	g	l	n	r	u	w
"i"	p"	q	a	d	i	l	m	s	t	w
"i"	q"	q	a	d	i	j	m	p	u	w
"i"	r"	q	c	e	i	l	n	s	t	w
"i"	s"	q	c	e	h	k	n	s	t	w
"i"	t"	q	a	e	h	l	o	s	t	y
"i"	u"	q	a	e	g	j	m	p	t	w
"i"	v"	q	a	e	i	j	m	p	t	w
"i"	w"	q	a	e	g	j	m	p	t	w
"i"	x"	q	a	e	g	j	m	p	t	w
"i"	y"	q	a	e	g	j	m	p	t	w
"i"	z"	q	a	e	i	j	o	p	t	w
"i"	"	q	a	d	g	j	m	p	t	w
"j"	a"	q	c	d	g	j	n	p	t	w
"j"	b"	q	a	d	g	j	m	p	t	w
"j"	c"	q	a	d	g	j	m	p	t	w
"j"	d"	q	a	d	g	j	m	p	t	w
"j"	e"	q	a	d	g	j	m	r	t	w
"j"	f"	q	a	d	g	j	m	p	t	w
"j"	g"	q	a	d	g	j	m	p	t	w
"j"	h"	q	a	d	g	j	m	p	t	w
"j"	i"	q	a	d	g	j	m	p	t	w
"j"	j"	q	a	d	g	j	m	p	t	w
"j"	k"	q	a	d	g	j	m	p	t	w
"j"	l"	q	a	d	g	j	m	p	t	w
"j"	m"	q	a	d	g	j	m	p	t	w
"j"	n"	q	a	d	g	j	m	p	t	w
"j"	o"	q	a	b	e	j	m	r	t	w
"j"	p"	q	a	d	g	j	m	p	t	w

FIG. 10G

"j	q"	1	2	3	4	5	6	7	8	9
"j	r"	q	a	d	g	j	m	p	t	w
"j	s"	q	a	d	g	j	m	p	t	w
"j	t"	q	a	d	g	j	m	p	t	w
"j	u"	q	a	d	g	l	m	s	t	w
"j	v"	q	a	d	g	j	m	p	t	w
"j	w"	q	a	d	g	j	m	p	t	w
"j	x"	q	a	d	g	j	m	p	t	w
"j	y"	q	a	d	g	j	m	p	t	w
"j	z"	q	a	d	g	j	m	p	t	w
"k	a"	q	a	d	g	j	n	s	t	y
"k	b"	q	a	d	g	j	o	p	t	w
"k	c"	q	a	d	g	j	m	p	t	w
"k	d"	q	a	d	g	l	o	p	t	w
"k	e"	q	a	d	g	j	n	s	t	y
"k	f"	q	a	d	g	j	m	p	t	w
"k	g"	q	a	d	g	j	m	r	t	w
"k	h"	q	a	d	g	j	m	p	t	w
"k	i"	q	a	d	g	l	n	p	t	w
"k	j"	q	a	d	g	j	m	p	t	w
"k	k"	q	a	d	g	j	o	p	t	w
"k	l"	q	a	d	g	j	m	p	t	y
"k	m"	q	a	d	g	j	m	p	t	w
"k	n"	q	a	d	g	l	o	p	t	w
"k	o"	q	a	d	g	j	n	p	t	w
"k	p"	q	a	d	g	j	m	p	t	w
"k	q"	q	a	d	g	j	m	p	t	w
"k	r"	q	a	d	g	j	m	p	t	w
"k	s"	q	a	d	g	j	m	p	t	w
"k	t"	q	a	d	g	j	m	p	t	w
"k	u"	q	a	d	g	j	m	p	t	w
"k	v"	q	a	d	g	j	m	p	t	w
"k	w"	q	a	d	g	j	m	p	t	w
"k	x"	q	a	d	g	j	m	p	t	w
"k	y"	q	a	d	g	j	m	p	t	w
"k	z"	q	a	d	g	j	m	p	t	w

FIG. 10H

"k	"	1	2	3	4	5	6	7	8	9
"l	"a"	q	a	d	g	j	m	p	t	w
"l	"b"	q	b	d	i	j	n	r	t	y
"l	"c"	q	a	d	g	j	o	p	t	w
"l	"d"	q	a	d	i	j	o	p	t	w
"l	"e"	q	a	d	g	l	n	s	t	w
"l	"f"	q	a	d	g	j	m	s	t	x
"l	"g"	q	a	d	g	j	m	p	t	w
"l	"h"	q	a	d	g	j	m	p	t	w
"l	"i"	z	a	d	g	k	n	s	t	w
"l	"j"	q	a	d	g	j	m	p	t	w
"l	"k"	q	a	d	i	j	m	s	t	w
"l	"l"	q	a	d	i	j	o	s	u	y
"l	"m"	q	a	d	g	j	o	s	t	w
"l	"n"	q	a	d	g	j	o	p	t	w
"l	"o"	q	a	d	g	j	o	s	t	w
"l	"p"	q	a	d	h	j	m	s	t	w
"l	"q"	q	a	d	g	j	m	p	t	w
"l	"r"	q	a	d	g	j	m	p	t	w
"l	"s"	q	a	e	i	j	o	p	t	w
"l	"t"	q	a	e	i	j	o	s	t	w
"l	"u"	q	a	e	i	l	m	s	t	w
"l	"v"	q	a	e	i	j	m	p	t	w
"l	"w"	q	a	e	g	j	m	p	t	w
"l	"x"	q	a	d	g	j	m	p	t	w
"l	"y"	z	a	d	i	j	n	s	t	w
"l	"z"	q	a	d	g	j	m	p	t	w
"m	"a"	q	a	d	i	k	m	r	t	y
"m	"b"	q	a	d	i	l	m	p	t	y
"m	"c"	q	a	d	i	j	o	p	t	w
"m	"d"	q	a	d	g	l	n	s	t	w
"m	"e"	q	a	d	g	j	o	p	t	w
"m	"f"	q	a	d	g	j	o	p	t	w
"m	"g"	q	a	d	g	j	m	p	t	w
"m	"h"	z	a	d	g	j	m	p	t	w
"m	"i"	z	c	d	g	l	n	s	t	x

FIG. 10I

"m	j"	1	2	3	4	5	6	7	8	9
"m	k"	q	a	d	g	j	m	p	t	w
"m	l"	q	a	d	g	j	m	p	t	w
"m	m"	q	a	e	g	j	m	p	t	w
"m	n"	q	a	e	g	j	m	p	t	w
"m	o"	q	a	d	g	j	m	p	r	w
"m	p"	q	b	d	g	j	n	r	v	w
"m	q"	q	a	e	g	l	o	r	t	w
"m	r"	q	a	d	g	j	m	p	t	w
"m	s"	q	a	e	g	j	m	p	t	w
"m	t"	q	a	d	g	j	m	p	t	w
"m	u"	q	a	d	g	l	n	s	t	w
"m	v"	q	a	d	g	j	m	p	t	w
"m	w"	q	a	d	g	j	m	p	t	w
"m	x"	q	a	d	g	j	m	p	s	w
"m	y"	q	a	d	g	j	m	p	s	w
"m	z"	q	a	d	g	j	m	p	r	w
"m	"	q	a	d	g	j	m	p	r	w
"n	a"	q	b	d	g	l	m	p	r	w
"n	b"	q	a	d	g	l	o	r	s	y
"n	c"	q	a	e	g	l	o	s	t	y
"n	d"	q	a	e	g	l	o	s	t	w
"n	e"	q	a	e	g	l	o	s	u	w
"n	f"	q	a	e	g	l	o	s	u	w
"n	g"	q	a	e	g	j	m	p	t	w
"n	h"	q	a	d	g	l	n	s	t	x
"n	i"	z	a	d	g	j	n	s	u	w
"n	j"	q	a	e	g	j	n	s	u	w
"n	k"	q	a	e	g	j	o	p	t	y
"n	l"	q	a	e	g	j	o	p	t	w
"n	m"	q	a	e	g	j	o	p	t	w
"n	n"	q	a	e	g	j	o	p	r	y
"n	o"	q	b	d	g	j	o	p	r	w
"n	p"	q	a	d	g	j	o	p	r	w
"n	q"	q	a	d	g	j	o	p	r	w
"n	r"	q	a	d	g	j	o	p	r	w
"n	s"	q	a	e	g	l	m	p	t	w

FIG. 10J

"	"	1	2	3	4	5	6	7	8	9
"n	t"	q	a	e	i	l	o	s	u	y
"n	u"	q	a	f	g	j	m	s	t	x
"n	v"	q	a	e	i	j	o	s	t	w
"n	w"	q	a	e	g	j	m	p	t	w
"n	x"	q	a	d	g	j	m	p	t	w
"n	y"	q	b	d	h	j	m	p	t	w
"n	z"	q	a	d	g	j	m	p	t	w
"n	"	q	a	d	g	j	m	p	t	w
"o	a"	q	b	d	g	l	m	s	t	w
"o	b"	q	a	e	i	l	o	s	t	w
"o	c"	q	a	e	h	k	o	s	u	w
"o	d"	q	a	e	i	l	o	s	u	y
"o	e"	q	a	d	g	j	m	s	t	w
"o	f"	q	a	f	g	j	m	p	t	w
"o	g"	q	a	e	i	j	o	n	r	y
"o	h"	q	a	d	g	j	n	p	t	w
"o	i"	q	c	d	g	l	n	p	t	w
"o	j"	q	a	e	g	j	m	p	t	w
"o	k"	q	a	e	i	j	m	s	t	w
"o	l"	q	b	d	i	l	o	s	u	w
"o	m"	q	a	e	i	l	m	s	t	w
"o	n"	q	a	e	g	l	n	s	t	y
"o	o"	q	a	d	g	k	n	s	t	w
"o	p"	q	a	e	h	l	m	p	t	y
"o	q"	q	a	d	g	j	m	p	t	w
"o	r"	q	a	e	g	k	m	r	t	w
"o	s"	q	a	e	i	j	o	s	t	y
"o	t"	q	c	e	h	j	o	s	t	y
"o	u"	q	a	e	g	l	n	r	t	y
"o	v"	q	a	e	i	j	m	p	t	w
"o	w"	q	a	e	i	j	n	s	t	w
"o	x"	q	a	d	i	j	m	p	t	w
"o	y"	q	a	d	g	j	m	p	t	w
"o	z"	q	a	d	g	j	m	p	t	w
"o	"	q	a	d	g	j	m	p	t	w
"p	a"	q	a	d	g	l	n	r	t	y
"p	b"	q	a	e	g	j	m	p	t	w

FIG. 10K

		1	2	3	4	5	6	7	8	9
"p	c	q	a	d	g	j	m	s	t	w
"p	d	q	a	d	g	j	m	p	t	w
"p	e	q	a	d	g	j	n	r	t	w
"p	f	q	a	d	g	j	m	r	t	w
"p	g	q	a	d	e	j	m	p	t	w
"p	h	q	a	d	e	j	o	s	t	y
"p	i	q	c	d	e	j	n	r	t	x
"p	j	q	a	d	e	j	m	p	t	w
"p	k	q	a	d	e	j	m	p	t	w
"p	l	q	a	e	e	j	o	p	u	y
"p	m	q	a	e	d	j	m	p	t	w
"p	n	q	a	d	e	j	m	p	t	w
"p	o	q	c	d	e	j	n	r	t	w
"p	p	q	a	e	e	j	o	p	t	y
"p	q	q	a	e	d	j	m	p	t	w
"p	r	q	a	e	d	j	o	p	t	y
"p	s	q	a	e	d	j	o	p	t	w
"p	t	q	c	d	e	j	o	s	t	w
"p	u	q	b	d	d	j	m	p	t	w
"p	v	q	a	d	d	j	m	p	t	w
"p	w	q	a	d	d	j	m	p	t	w
"p	x	q	a	d	d	j	m	p	t	w
"p	y	q	a	d	d	j	m	p	t	w
"p	z	q	a	d	d	j	m	p	t	w
"q	.	q	a	d	d	j	m	p	t	w
"q	a	q	a	d	d	j	m	p	t	w
"q	b	q	a	d	d	j	m	p	t	w
"q	c	q	a	d	d	j	m	p	t	w
"q	d	q	a	d	d	j	m	p	t	w
"q	e	q	a	d	d	j	m	p	t	w
"q	f	q	a	d	d	j	m	p	t	w
"q	g	q	a	d	d	j	m	p	t	w
"q	h	q	a	d	d	j	m	p	t	w
"q	i	q	a	d	d	j	m	p	t	w
"q	j	q	a	d	d	j	m	p	t	w
"q	k	q	a	d	d	j	m	p	t	w
"q	l	q	a	d	d	j	m	p	t	w

FIG. 10L

"q	m	1	2	3	4	5	6	7	8	9
"q	n	q	a	d	g	j	m	p	t	w
"q	o	q	a	d	g	j	m	p	t	w
"q	p	q	a	d	g	j	m	p	t	w
"q	q	q	a	d	g	j	m	p	t	w
"q	r	q	a	d	g	j	m	p	t	w
"q	s	q	a	d	g	j	m	p	t	w
"q	t	q	a	d	g	j	m	p	t	w
"q	u	q	a	d	g	j	m	p	t	w
"q	v	q	a	d	g	j	m	p	t	w
"q	w	q	a	d	g	j	m	p	t	w
"q	x	q	a	d	g	j	m	p	t	w
"q	y	q	a	d	g	j	m	p	t	w
"q	z	q	a	d	g	j	m	p	t	w
"r	"	q	a	d	g	j	m	p	t	w
"r	a	q	c	d	g	l	n	p	t	w
"r	b	q	a	d	g	l	o	p	t	w
"r	c	q	a	d	g	l	m	p	s	v
"r	d	q	a	e	g	l	n	s	v	w
"r	e	q	a	e	g	j	o	s	t	w
"r	f	q	a	e	g	j	m	s	u	w
"r	g	q	a	e	g	j	o	p	t	w
"r	h	q	a	e	g	l	n	p	t	w
"r	i	z	c	e	g	l	m	p	t	w
"r	j	q	a	e	g	l	m	s	t	y
"r	k	q	a	e	i	l	o	p	t	y
"r	l	q	a	e	i	j	m	s	t	w
"r	m	q	a	e	i	j	o	s	t	w
"r	n	q	a	e	g	l	m	p	t	w
"r	o	q	b	d	g	j	o	r	t	w
"r	p	q	a	d	g	j	m	p	t	w
"r	q	q	a	d	g	j	o	p	t	y
"r	r	q	a	e	i	j	o	p	t	w
"r	s	q	a	e	i	j	n	s	u	w
"r	t	z	a	e	h	l	n	s	t	w
"r	u	q	c	e	g	l	m	s	t	w
"r	v	q	a	e	i	j	m	p	t	w

FIG. 10M

"r	w"	1	2	3	4	5	6	7	8	9
"r	x"	q	a	d	i	j	o	p	t	w
"r	y"	q	a	d	g	j	m	p	t	w
"r	z"	q	b	d	g	j	m	p	t	w
"s	"	q	a	d	g	j	m	p	t	w
"s	a"	q	b	d	g	l	m	p	v	y
"s	b"	q	a	d	g	j	m	p	t	w
"s	c"	q	a	d	h	l	o	r	u	w
"s	d"	q	a	d	g	j	n	p	t	w
"s	e"	q	c	e	g	l	n	r	t	y
"s	f"	q	a	e	g	j	m	p	u	w
"s	g"	q	a	e	g	j	m	r	t	w
"s	h"	q	a	e	g	l	o	r	t	w
"s	i"	z	c	d	g	l	n	s	t	x
"s	j"	q	a	d	g	j	m	p	t	w
"s	k"	q	a	e	i	j	m	p	t	w
"s	l"	q	a	e	i	j	o	p	t	y
"s	m"	q	a	e	i	j	o	s	t	w
"s	n"	q	a	e	g	j	m	p	t	w
"s	o"	q	c	f	g	l	m	r	u	w
"s	p"	q	a	e	g	l	o	r	t	w
"s	q"	q	a	d	g	l	m	p	u	w
"s	r"	q	c	d	g	j	m	p	t	w
"s	s"	q	a	e	i	j	o	p	u	w
"s	t"	q	a	e	i	l	o	r	u	y
"s	u"	q	b	e	g	l	m	p	t	w
"s	v"	q	a	e	g	j	m	p	t	w
"s	w"	q	a	e	i	j	o	p	t	w
"s	x"	q	a	d	g	j	m	p	t	w
"s	y"	q	a	d	g	j	m	s	t	w
"s	z"	q	a	d	g	j	m	p	t	w
"t	"	q	a	d	g	j	m	p	t	w
"t	a"	q	c	f	i	l	n	r	t	y
"t	b"	q	a	d	g	j	m	p	t	w
"t	c"	z	a	d	h	j	o	p	t	w
"t	d"	q	a	d	g	j	m	p	t	w
"t	e"	q	c	d	g	l	m	r	u	x

FIG. 10N

"	"	1	2	3	4	5	6	7	8	9
"t	f	q	a	d	i	j	o	p	t	w
"t	g	q	a	e	i	j	o	p	t	w
"t	h	q	a	e	i	j	o	p	r	w
"t	i	z	c	e	g	l	o	r	u	w
"t	j	q	a	d	g	j	o	s	v	w
"t	k	q	a	e	g	j	m	p	t	w
"t	l	q	a	e	i	j	m	p	t	w
"t	m	q	a	e	g	l	o	p	t	y
"t	n	q	a	e	g	l	m	p	t	w
"t	o	q	a	c	g	l	m	r	u	w
"t	p	q	c	d	g	j	m	p	u	w
"t	q	q	a	d	g	j	m	p	u	w
"t	r	q	a	e	i	j	o	p	u	y
"t	s	q	c	e	i	j	o	p	u	w
"t	t	q	a	e	i	l	o	p	t	y
"t	u	q	a	f	i	j	n	r	t	w
"t	v	q	a	d	i	j	m	p	t	w
"t	w	q	a	e	g	j	o	p	t	w
"t	x	q	a	d	g	l	m	p	t	w
"t	y	q	a	d	g	l	m	p	t	w
"t	z	q	a	d	g	j	m	p	t	w
"t	"	q	a	d	g	l	m	p	t	w
"u	a	q	a	d	g	l	n	r	t	w
"u	b	q	a	e	i	j	m	s	t	w
"u	c	q	c	e	h	k	m	p	t	w
"u	d	q	a	e	i	j	o	s	t	y
"u	e	q	a	d	g	j	n	s	t	w
"u	f	q	a	f	g	j	m	s	t	w
"u	g	q	a	e	h	j	m	s	t	w
"u	h	q	a	d	g	j	m	p	t	w
"u	i	q	c	d	g	l	n	r	t	w
"u	j	q	a	d	g	j	m	p	t	w
"u	k	q	a	d	g	l	m	p	t	w
"u	l	q	a	d	g	l	m	p	t	y
"u	m	q	a	e	i	j	m	p	t	w
"u	n	q	c	d	i	l	n	s	t	w
"u	o	q	a	d	g	j	m	p	t	w

FIG. 100

"	"	1	2	3	4	5	6	7	8	9
"u	p"	q	a	d	i	l	o	p	t	y
"u	q"	q	a	d	g	j	m	p	t	w
"u	r"	q	a	d	i	l	n	s	t	y
"u	s"	q	a	e	i	l	m	s	t	y
"u	t"	q	a	e	i	l	o	s	t	w
"u	u"	q	a	d	g	j	m	p	t	w
"u	v"	q	a	d	g	j	m	p	t	w
"u	w"	q	a	d	g	j	m	p	t	w
"u	x"	q	a	d	g	j	m	p	t	w
"u	y"	q	a	e	g	j	m	s	t	w
"u	z"	q	a	d	g	j	m	p	t	w
"v	a"	q	a	d	g	j	m	p	t	w
"v	b"	q	a	d	g	j	m	p	t	w
"v	c"	q	a	d	g	j	m	s	t	w
"v	d"	q	a	d	g	j	m	p	t	w
"v	e"	q	a	d	g	j	n	r	t	w
"v	f"	q	a	d	g	j	m	p	t	w
"v	g"	q	a	d	g	j	m	p	t	w
"v	h"	q	a	d	g	j	m	p	t	w
"v	i"	q	a	e	g	j	n	s	t	w
"v	j"	q	a	d	g	j	m	p	t	w
"v	k"	q	a	d	g	j	m	p	t	w
"v	l"	q	a	d	g	j	m	s	t	w
"v	m"	q	a	d	g	j	m	p	t	w
"v	n"	q	a	d	g	j	m	p	t	w
"v	o"	q	a	d	i	l	o	r	t	w
"v	p"	q	a	d	g	j	m	p	t	w
"v	q"	q	a	d	g	j	m	p	t	w
"v	r"	q	a	d	g	j	m	p	t	w
"v	s"	q	a	d	g	j	m	p	t	w
"v	t"	q	a	d	g	j	m	p	t	w
"v	u"	q	a	d	g	j	m	p	t	w
"v	v"	q	a	d	g	j	m	p	t	w
"v	w"	q	a	d	g	j	m	p	t	w
"v	x"	q	a	d	g	j	m	p	t	w
"v	y"	q	a	d	g	j	m	p	t	w

FIG. 10P

		1	2	3	4	5	6	7	8	9
"v	z"	q	a	d	g	j	m	p	t	w
"v	"	q	a	d	g	j	m	p	t	w
"w	a"	q	b	d	g	j	n	r	v	y
"w	b"	q	a	d	g	j	m	p	t	w
"w	c"	q	a	e	g	j	m	p	t	w
"w	d"	q	a	f	g	j	m	p	t	w
"w	e"	q	b	e	i	j	n	r	v	w
"w	f"	q	a	d	g	j	m	p	t	w
"w	g"	q	a	d	g	j	m	p	t	w
"w	h"	q	a	e	i	j	o	p	t	y
"w	i"	q	c	d	g	j	n	s	t	w
"w	j"	q	a	d	g	j	m	p	t	w
"w	k"	q	a	d	g	j	m	p	t	w
"w	l"	q	a	d	g	j	m	p	t	w
"w	m"	q	a	d	g	j	m	p	t	w
"w	n"	q	a	d	g	j	m	s	t	w
"w	o"	q	a	d	g	j	n	r	u	w
"w	p"	q	a	d	g	j	m	p	t	w
"w	q"	q	a	d	g	j	m	p	t	w
"w	r"	q	a	d	i	j	o	p	t	w
"w	s"	q	a	e	g	j	m	p	t	w
"w	t"	q	a	d	g	j	m	p	t	w
"w	u"	q	a	d	g	j	m	p	t	w
"w	v"	q	a	d	g	j	m	p	t	w
"w	w"	q	a	d	g	j	m	p	t	w
"w	x"	q	a	d	g	j	m	p	t	w
"w	y"	q	a	d	g	j	m	p	t	w
"w	z"	q	a	d	g	j	m	p	t	w
"x	"	q	a	d	g	j	m	p	t	w
"x	a"	q	c	d	g	j	m	p	t	w
"x	b"	q	a	d	g	j	m	p	t	w
"x	c"	q	a	e	h	j	m	p	t	w
"x	d"	q	a	d	g	j	m	p	t	w
"x	e"	q	a	d	g	j	m	s	t	x
"x	f"	q	a	d	g	j	m	p	t	w
"x	g"	q	a	d	g	j	m	p	t	w
"x	h"	q	a	d	g	j	m	p	t	w

FIG. 10Q

"	"	1	2	3	4	5	6	7	8	9
"x	i"	q	c	d	g	j	n	s	t	w
"x	j"	q	a	d	g	j	m	p	t	w
"x	k"	q	a	d	g	j	m	p	t	w
"x	l"	q	a	d	g	j	m	p	t	w
"x	m"	q	a	d	g	j	m	p	t	w
"x	n"	q	a	d	g	j	m	p	t	w
"x	o"	q	a	d	g	j	m	p	t	w
"x	p"	q	a	e	g	j	o	p	t	w
"x	q"	q	a	d	g	j	m	p	t	w
"x	r"	q	a	d	g	j	m	p	t	w
"x	s"	q	a	e	g	j	m	p	t	w
"x	t"	q	a	e	g	j	m	r	u	w
"x	u"	q	a	d	g	j	m	p	t	w
"x	v"	q	a	d	g	j	m	p	t	w
"x	w"	q	a	d	g	j	m	p	t	w
"x	x"	q	a	d	g	j	m	p	t	w
"x	y"	z	a	d	g	j	m	p	t	w
"x	z"	q	a	d	g	j	m	p	t	w
"y	a"	q	a	d	g	j	m	p	t	w
"y	b"	q	a	e	g	j	o	p	t	w
"y	c"	q	a	d	g	j	o	p	t	w
"y	d"	q	a	d	g	j	m	s	t	w
"y	e"	q	a	d	g	j	m	p	t	w
"y	f"	q	a	d	g	j	m	p	t	w
"y	g"	q	a	d	g	j	o	p	t	w
"y	h"	q	a	d	g	j	n	p	t	w
"y	i"	q	c	d	g	j	m	p	t	w
"y	j"	q	a	d	g	j	m	p	t	w
"y	k"	q	a	e	g	j	m	p	t	w
"y	l"	q	a	e	g	j	o	p	t	w
"y	m"	q	a	d	g	j	m	p	t	w
"y	n"	q	c	d	g	j	n	r	u	x
"y	o"	q	a	e	g	j	m	p	t	w
"y	p"	q	a	d	g	j	m	p	t	w
"y	q"	q	a	d	g	j	m	p	t	w
"y	r"	q	a	d	g	j	m	p	t	w

FIG. 10R

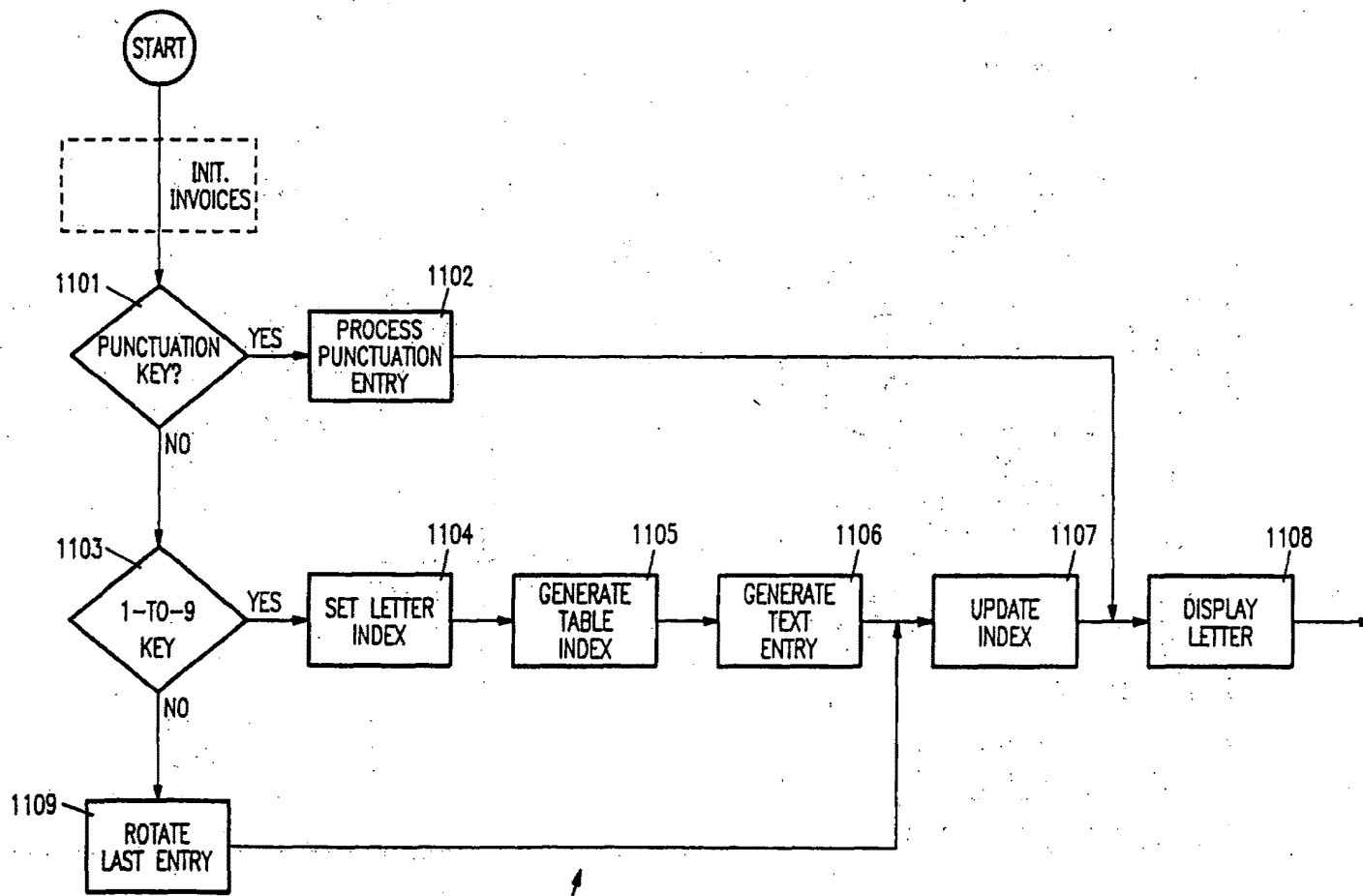
"	-	1	2	3	4	5	6	7	8	9
"y	s"	q	c	e	i	j	m	p	t	w
"y	t"	q	a	e	h	j	m	p	t	w
"y	u"	q	a	d	g	k	m	r	t	w
"y	v"	q	a	d	g	j	m	p	t	w
"y	w"	q	a	d	h	j	m	p	t	w
"y	x"	q	a	d	g	j	m	p	t	w
"y	y"	q	a	d	g	j	m	p	t	w
"y	z"	q	a	e	g	j	m	p	t	w
"y	"	q	a	d	g	j	m	p	t	w
"z	a"	q	a	d	g	j	m	p	t	w
"z	b"	q	a	d	g	j	m	p	t	w
"z	c"	q	a	d	g	j	m	p	t	w
"z	d"	q	a	d	g	j	m	p	t	w
"z	e"	q	a	d	g	j	n	r	t	w
"z	f"	q	a	d	g	j	m	p	t	w
"z	g"	q	a	d	g	j	m	p	t	w
"z	h"	q	a	d	g	j	m	p	t	w
"z	i"	q	a	d	g	j	n	p	t	w
"z	j"	q	a	d	g	j	m	p	t	w
"z	k"	q	a	d	g	j	m	p	t	w
"z	l"	q	a	d	g	j	m	p	t	w
"z	m"	q	a	d	g	j	m	p	t	w
"z	n"	q	a	e	g	j	m	p	t	w
"z	o"	q	a	d	g	j	n	p	t	w
"z	p"	q	a	d	g	j	m	p	t	w
"z	q"	q	a	d	g	j	m	p	t	w
"z	r"	q	a	d	g	j	m	p	t	w
"z	s"	q	a	d	g	j	m	p	t	w
"z	t"	q	a	d	g	j	m	p	t	w
"z	u"	q	a	d	g	j	m	p	t	w
"z	v"	q	a	d	g	j	m	p	t	w
"z	w"	q	a	d	g	j	m	p	t	w
"z	x"	q	a	d	g	j	m	p	t	w
"z	y"	q	a	d	g	j	m	p	t	w
"z	z"	q	a	d	g	j	m	p	t	w
"z	"	q	a	d	g	j	m	p	t	w
"	a"	q	c	d	i	l	n	r	t	w

FIG. 10S

"	-	1	2	3	4	5	6	7	8	9
"	b"	q	a	e	i	l	o	r	u	y
"	c"	q	a	e	h	l	o	r	u	y
"	d"	q	a	e	i	j	o	r	u	w
"	e"	q	a	d	i	l	n	s	v	x
"	f"	q	a	e	i	l	o	r	u	w
"	g"	q	a	e	i	l	o	r	u	w
"	h"	q	a	e	i	j	o	p	t	w
"	i"	q	c	f	g	j	n	s	t	w
"	j"	q	a	e	g	j	o	p	u	w
"	k"	q	b	e	i	j	n	p	t	w
"	l"	q	a	e	i	l	o	p	u	y
"	m"	q	a	e	i	k	o	s	u	y
"	n"	q	a	e	i	j	o	p	u	w
"	o"	z	b	f	g	k	n	r	u	w
"	p"	z	a	e	h	l	o	r	u	w
"	q"	z	a	d	g	j	m	p	u	w
"	r"	q	a	e	i	j	o	s	u	w
"	s"	q	c	e	h	l	o	p	t	y
"	t"	q	a	e	h	j	o	r	u	w
"	u"	q	a	d	i	k	n	s	t	w
"	v"	q	a	e	i	l	o	p	t	w
"	w"	q	a	e	i	j	o	r	t	w
"	x"	q	a	d	g	j	m	p	t	y
"	y"	q	a	e	i	j	o	p	u	w
"	z"	q	a	f	i	l	o	s	t	w

FIG. 10T

101



1100

FIG. 11

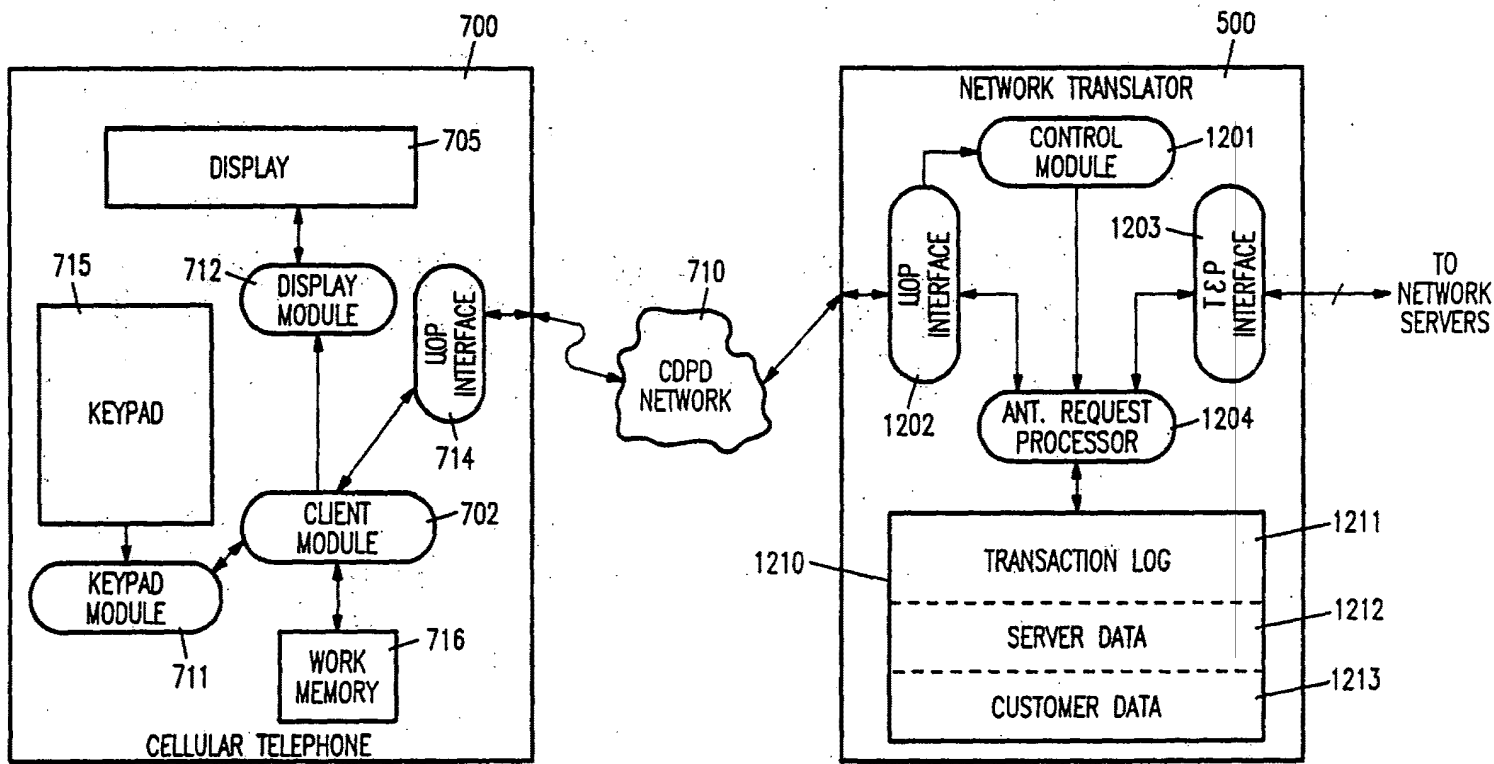


FIG. 12

102

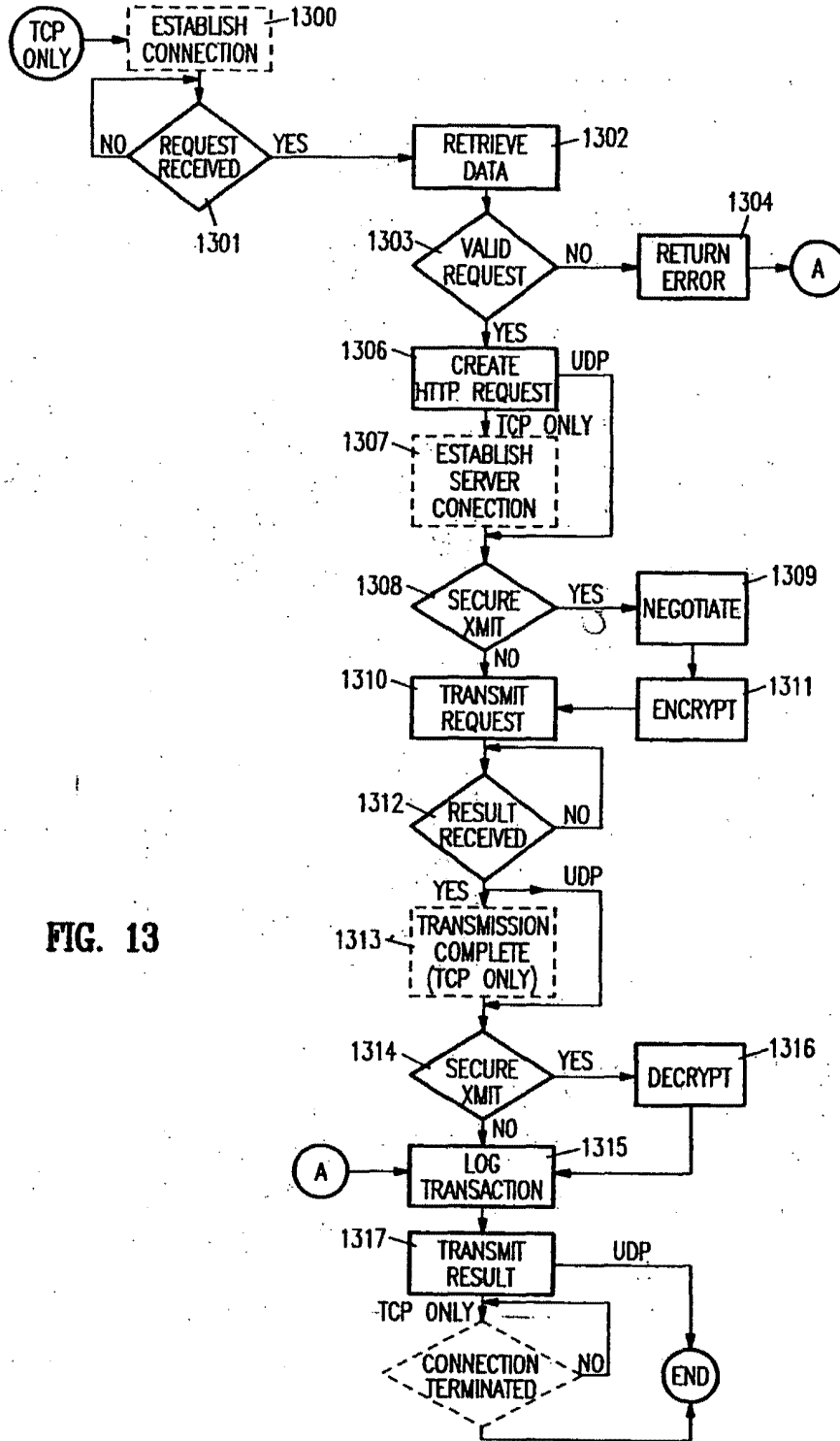


FIG. 13

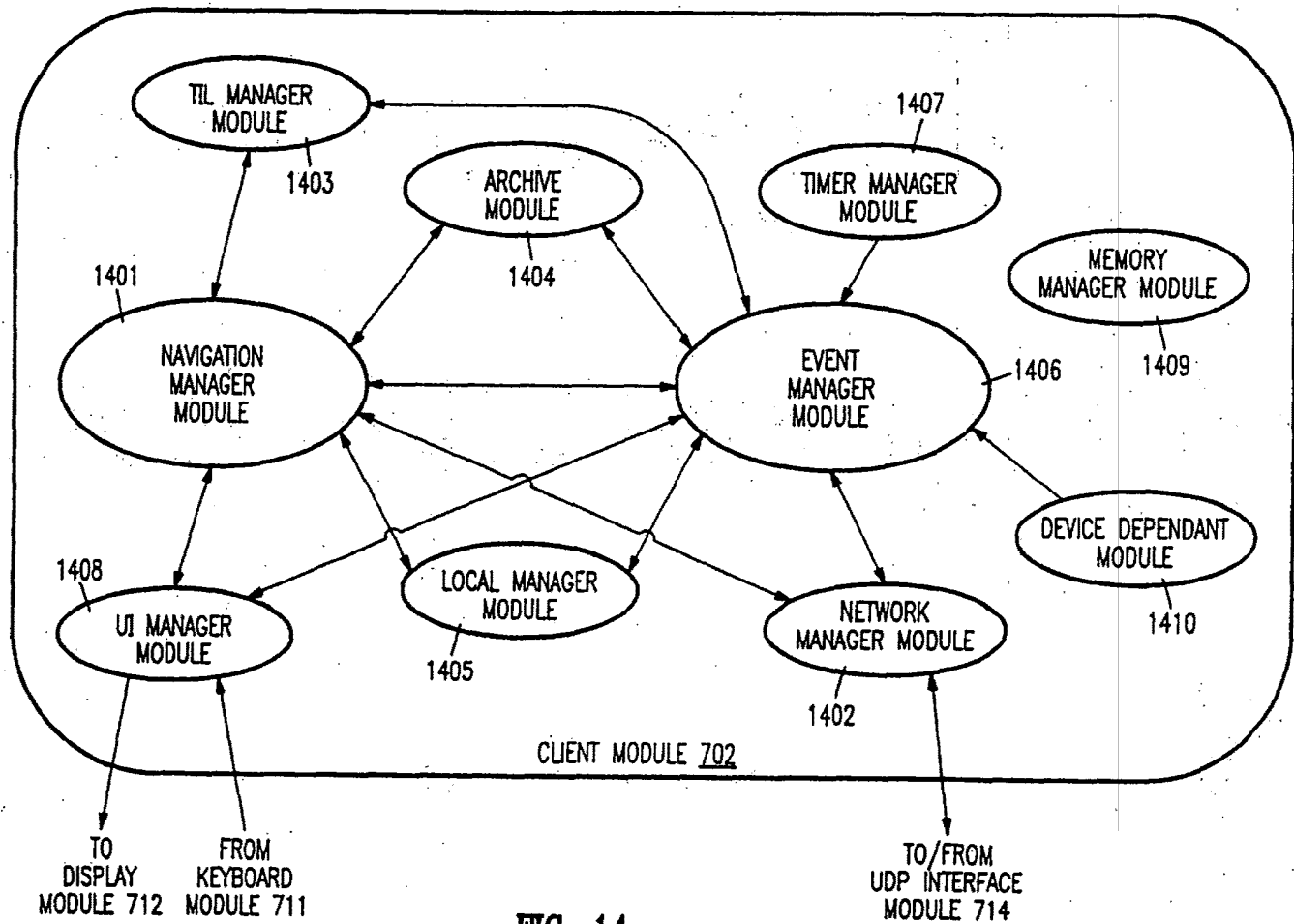


FIG. 14

104

EP 0 779 759 A2

(19)



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11)

EP 0 779 759 A3

(12)

EUROPEAN PATENT APPLICATION

(88) Date of publication A3:
17.11.1999 Bulletin 1999/46

(51) Int. Cl.⁶: H04Q 7/24, H04L 12/28

(43) Date of publication A2:
18.06.1997 Bulletin 1997/25

(21) Application number: 96309006.3

(22) Date of filing: 11.12.1996

(84) Designated Contracting States:
AT BE CH DE DK ES FI FR GB GR IE IT LI LU MC
NL PT SE

(72) Inventor: Rossmann, Alain
Palo Alto, CA 94303 (US)

(30) Priority: 11.12.1995 US 570210
11.12.1995 US 570384

(74) Representative:
Freeman, Jacqueline Carol
W.P. THOMPSON & CO.
Celcon House
289-293 High Holborn
London WC1V 7HU (GB)

(71) Applicant: Unwired Planet, Inc.
Redwood City, CA 94065 (US)

(54) **A method and architecture for an interactive two-way data communication network**

(57) A two-way data communication device such as a data ready cellular telephone, a two-way pager, or a telephone communicates via a two-way data communication network with a server computer on a computer network that has an interface to the two-way data communication network, i.e. is coupled to the two-way data communication network. For example, the computer network can be a corporate wide area network, a corporate local area network, the Internet, or any combination of computer networks. The two-way data communication device utilizes a client module to transmit message including a resource selector chosen by the user to a server on a server computer on the computer network. The server processes the message and transmits a response over the two-way data communication network to the client module. The client module interprets the response and presents the response to the user via a structured user interface. Alternatively, the user transmits a request that directs the server to transmit the response to the request to another location or to another user.

EP 0 779 759 A3



European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 96 30 9006

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
X	LILJEBERG M ET AL: "OPTIMIZING WORLD-WIDE WEB FOR WEAKLY CONNECTED MOBILE WORKSTATIONS:AN INDIRECT APPROACH" INTERNATIONAL WORKSHOP ON SERVICES IN DISTRIBUTED AND NETWORKED ENVIRONMENTS, 5 June 1995 (1995-06-05), pages 132-139, XP000764774. * the whole document *	1-11, 25-29	H0407/24 H04L12/28
A	KAASHOEK M F ET AL: "DYNAMIC DOCUMENTS: MOBILE WIRELESS ACCESS TO THE WWW" PROCEEDINGS, WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS, 8 December 1994 (1994-12-08), pages 179-184, XP002016896 * the whole document *	1-11, 23-29	
X	WO 93 16550 A (BELL ATLANTIC NETWORK SERVICES) 19 August 1993 (1993-08-19) * page 10, line 15 - line 22 * * page 11, line 19 - page 12, line 2 * * page 17, line 6 - page 18, line 14; figures 6,7 *	36 1,2	
A	GESSLER S ET AL: "PDAs as mobile WWW browsers" COMPUTER NETWORKS AND ISDN SYSTEMS, vol. 28, no. 1, 1 December 1995 (1995-12-01), page 53-59 XP004001210 ISSN: 0169-7552 * the whole document *	36-39	TECHNICAL FIELDS SEARCHED (Int.Cl.8) H04Q H04L G06F
-/--			
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 23 September 1999	Examiner Gerling, J.C.J.
CATEGORY OF CITED DOCUMENTS X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document		T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document	

EPO FORM 1603 03 02 (P/IntCl.6)



European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 96 30 9006

DOCUMENTS CONSIDERED TO BE RELEVANT			
Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.6)
A	VOELKER G M ET AL: "MOBISAIC: AN INFORMATION SYSTEM FOR A MOBILE WIRELESS COMPUTING ENVIRONMENT" PROCEEDINGS, WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS, vol. 5, no. 10, 1 January 1995 (1995-01-01), pages 185-190, XP002062595		
A	US 5 128 672 A (KAEHLER EDWIN B) 7 July 1992 (1992-07-07)		
			TECHNICAL FIELDS SEARCHED (Int.Cl.6)
The present search report has been drawn up for all claims			
Place of search THE HAGUE		Date of completion of the search 23 September 1999	Examiner Gerling, J.C.J.
<p>CATEGORY OF CITED DOCUMENTS</p> <p>X : particularly relevant if taken alone Y : particularly relevant if combined with another document of the same category A : technological background O : non-written disclosure P : intermediate document</p> <p>T : theory or principle underlying the invention E : earlier patent document, but published on, or after the filing date D : document cited in the application L : document cited for other reasons & : member of the same patent family, corresponding document</p>			

EPO FORM 1503 (03/92) (PatCat)

**ANNEX TO THE EUROPEAN SEARCH REPORT
ON EUROPEAN PATENT APPLICATION NO.**

EP 96 30 9006

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on. The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

23-09-1999

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
WO 9316550 A	19-08-1993	AU 3601593 A	03-09-1993
US 5128672 A	07-07-1992	JP 5134797 A	01-06-1993

EPO FORM P0459

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82

XP-002114154



P.D. 06/04/98
P. Complete = 2

TechWeb News

Tech Search

Search news:

Search Home Advanced Search Search Help Web Search

EETIMES

April 06, 1998, Issue: 1001
Section: News

News

TechWeb Sites

W3C putting Compact HTML, HDML to test for Net access - Mobile markup languages face off

Yoshiko Hara

Tokyo - A compact version of the Hypertext Markup Language that targets portable Internet access will face off against the Handheld Device Markup Language here this week at a mobile-access workshop hosted by the World Wide Web Consortium (W3C)

Compact HTML was developed by Access Co. Ltd. (Tokyo), which last month submitted the language to W3C as a proposed standard. Fujitsu Ltd., Matsushita Electric Industrial Co. Ltd., Mitsubishi Electric Corp., NEC Corp. and Sony Corp. joined Access in making the proposal. The HTML subset would provide a compact browser to such portable gear as cellular phones, personal digital assistants and personal communicators.

The rival HDML was proposed to W3C last May by Unwired Planet, Alcatel Alsthom Recherche, AT&T, Gemplus, Mitsubishi Electric, Sun Microsystems and Tandem Computers.

Top Headlines
• Electronic Market Soars For Computer Users
• E-Trade.com Tries Digital Content
• Security Woes Continue For Microsoft
a more headlines

W3C has posted the Compact HTML proposal on its home page, where a "staff comment" suggests the consortium's preference for the HTML subset over HDML. The comment cites "another submission made [last year] to address similar [mobile access] requirements: the HDML Submission" and states that while HDML's backers "proposed a format similar to . . . HTML, the Compact HTML submission is closer to existing W3C recommendations."

Get CMPnet DELIVERED
Subscribe NOW - FREE!

Free E-mail

At Home Discovery
Download Now
Get
PrintCast

Compact HTML is designed so that a browser based on the language can be implemented in products with steep hardware restrictions. The browser could run on a system with 150 to 200 kbytes of RAM and 150 to 200 Mbytes of ROM. A processor with a relatively low 1 to 10 Mips of performance, similar to those commonly used for embedded applications, could control the Internet access. The system's display is also assumed to be small, featuring 50- x 30-dot resolution, 100- x 72-dot resolution, or 150- x 100-dot resolution with monochrome color and a single font. Four buttons-for forward cursor, backward cursor, select and cancel- will handle all basic operations.

Compact HTML is a subset of HTML 2.0, HTML 3.2 and HTML 4.0. To satisfy portable hardware restrictions, it will support GIF images but not such capabilities as JPEG images or tables, background color, or multiple character fonts and styles.

Officials backing the respective proposals are keenly aware of their competition.

"By contrast [to Compact HTML], HDML has no compatibility with HTML," said Tomihisa Kamada, executive vice president of Access and the leader of the language's development. "Compact HTML can use HTML infrastructure as it is. Simple pages can be viewed directly from portables based on Compact HTML, and home-page creators can write dedicated pages for mobile access easily as a part of HTML home pages."

Once W3C selects a standard, content providers will be able to prepare pages linked to conventional HTML home pages but accessible to mobile gear. "We

proposed the language to expand the market," said Kamada. Access has already developed a browser, called Compact NetFront, based on Compact HTML.



Mitsubishi supports both HDML and Compact HTML. The company is a shareholder of Unwired Planet and manufactures terminals based on HDML for AT&T. "Each format has different advantages; they can coexist in the market in different form of services," a Mitsubishi spokesman said.

An engineer for one of the proponents of Compact HTML said the language is the better of the announced alternatives.

NTT Mobile Communications Network Inc., the largest cellular carrier in Japan, contributed its cell-phone technology to Compact HTML. The company is considering a service based on the language. Such a move could provide vendors with an incentive to support it.

Copyright (c) 1998 CMP Media Inc.


Sponsored Supplements From CMPnet

 <p>Planet IT Plug your laptop into USB. Will it save headaches or create new ones?</p>	 <p>Small Biz Find out how affiliate programs can make your Web site money.</p>
--	--

- Find out what **knowledge management** can do for you.
- **Data mining** is one profitable use for all that data you store.
- Planet IT's newest tech center is about **Advanced IP services** — and it's hot!
- We take an in-depth look at Caldera's latest: **OpenLinux 2.2**.



For the best in Voice and Data solutions, visit Communications Central



BEST AVAILABLE COPY

order online from local restaurants

FOOD.COM mmmm

order online cooking & entertaining About Food.com contact us

Press Releases

Entertainment Industry Leader Rich Frank Takes Helm at Cybermeals, World's Largest Online Meal Ordering System

San Francisco, CA (September 15, 1998) - cybermeals (www.cybermeals.com), the world's largest online meal ordering system, today announced the appointment of Richard Frank as the new chairman, president and CEO of the two-year-old Internet start-up. Frank succeeds interim CEO Bill Jesse, who will remain as a member of the Board of Directors.

Frank has driven a wide range of initiatives in traditional and emerging sectors of the entertainment world throughout his career. His move from a foremost position in the entertainment industry to one in the fast-paced Internet commerce and food arena is another strong indication that e-commerce is not merely a trend, but a dominant industry segment attracting the attention and talent of top players. A fact reinforced by Frank's significant financial investment in cybermeals.

cybermeals was founded in 1996 and currently is available in major markets nationwide. The service gives Internet users the ability to locate restaurants in their own neighborhoods and order meals online as far as three months in advance. Currently, cybermeals has over 10,000 restaurants online. cybermeals has long-term, exclusive partnerships with the majority of restaurant delivery services throughout the U.S. and has been exclusively endorsed by a number of leading restaurant trade organizations. Additionally, cybermeals is the exclusive online meal ordering service for America Online, the nation's largest Internet service provider.

"In the rapidly-expanding world of Internet commerce, cybermeals is a dynamic young company with unlimited market opportunity. Based on recent industry estimates, the potential market for online food delivery is five times larger than that of the book and music industries. Leading the charge is a challenge I couldn't pass up and I am delighted to have the chance to build on the initial success of the cybermeals team," said Frank. "cybermeals is well-positioned to capitalize on the promise of the burgeoning Internet food arena with superb technology, outstanding strategic partnerships and a talented group of employees. I am confident that we will rapidly bring the company into its next phase of development and realize its extraordinary potential."

Frank was most recently chairman and CEO of Comcast Content and Communication (C3), a company he founded in 1995 with television/telecommunications giant Comcast Corporation. Prior to his position at C3, Frank served as chairman of Walt Disney Television and Telecommunications (1994 to 1995) and as

About Food.com

Company Information

[Press Room](#)
[Press Releases](#)
[In the News](#)
[Virtual Press Kit](#)
[PR Contacts](#)

Job Opportunities

[Contact Us](#)

1
president of The Walt Disney Studios (1985 to 1994). From 1977 to 1985, he served as both vice president and president of the Paramount Television Group of Paramount Pictures.

"Rich Frank is a proven leader with more than 25 years of senior management experience at major consumer-oriented corporations. We are very excited to have him on board both as Chairman and as a major investor," said Bill Jesse. "We immediately recognized Rich's talent for enhancing and building strong brands and his unusual ability to foster environments that generate products with enduring consumer appeal. His proven track record as a strong operational manager and business builder makes him a valuable asset to cybermeals."

"Rich Frank is an outstanding addition to cybermeals," said Jim Breyer, cybermeals Board Member and Managing Partner at Accel Partners. "Accel Partners has made a significant investment in the company and we are confident that under Rich's leadership, cybermeals is well positioned for success in this exciting market."

[Return to Press Release List](#)

[about us](#) [feedback](#) [jobs@food.com](#) [help](#) [privacy policy](#)
© 2000 Food.com

E

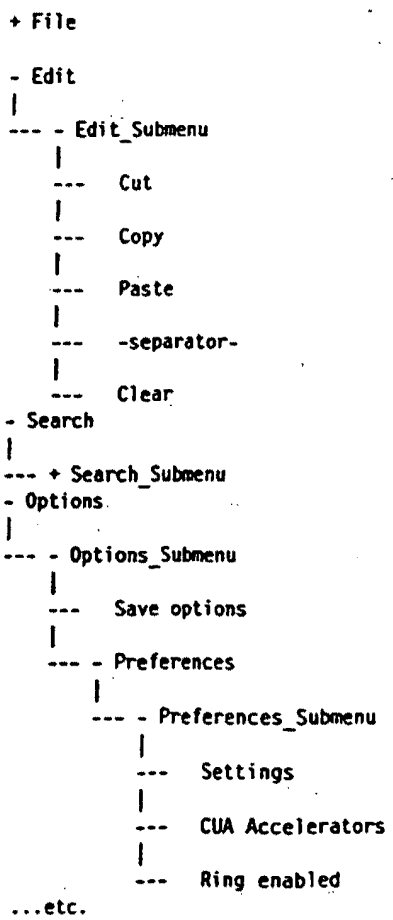
Graphic User Interface Builder Menu Construction using a Tree-View Container



p. 65-66

A method for arranging an ordered, hierarchical structure, such as a menu, using a common graphical interface is disclosed. The disclosed means for arranging a hierarchical structure uses a tree-view container. For the particular case of a menu illustrated below, each node in the tree would represent either a menu item or a sub menu. The following diagram (Menu X) illustrates the typical view presented to the user.

MENU X



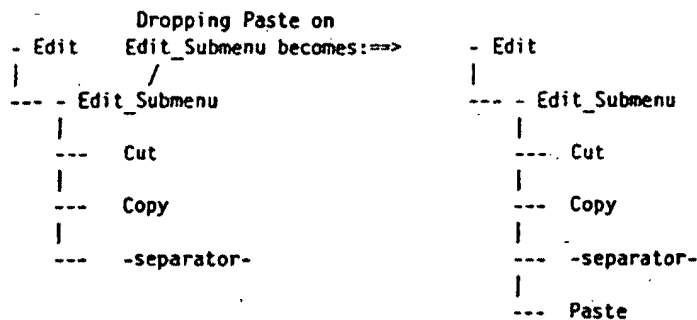
Graphic User Interface Builder Menu Construction using a Tree-View Container -- Continued

Pop-up menus on each node would be populated with items relevant to the items they represent. The container itself would represent the object (in this case a menu) as a whole, and would have a popup relevant to that object.

Manipulation of the nodes could be allowed through the popup-menus in order to provide keyboard support, however this system is intended primarily for use with a mouse, or some other pointing device.

The pointing device would be used to drag and drop the different nodes in order to rearrange them. Different keyboard combinations applied during the drag would indicate whether the intended action is a move or a copy. When a node is moved, or copied, all of its descendant nodes would be moved or copied along with it.

If an object is dropped directly on a node, the interpretation would be that the object is intended to be added as the last child of the target node.



If an object is dropped between two nodes at the same level, the interpretation is that the object should be inserted between the two nodes.



A similar method to that disclosed herein could be provided to allow the user to reorder the pages of a notebook control.

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**



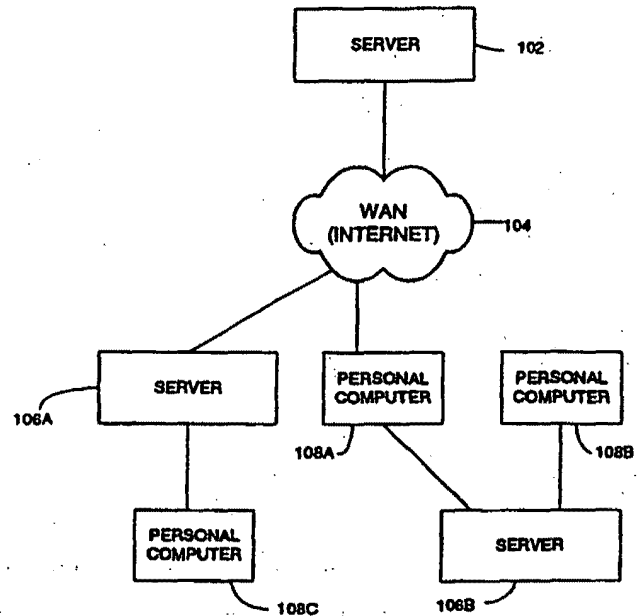
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification ⁶ : G06F 17/30</p>	<p>A2</p>	<p>(11) International Publication Number: WO 98/20434 (43) International Publication Date: 14 May 1998 (14.05.98)</p>
<p>(21) International Application Number: PCT/US97/19719 (22) International Filing Date: 30 October 1997 (30.10.97) (30) Priority Data: 08/745,899 7 November 1996 (07.11.96) US 08/762,289 9 December 1996 (09.12.96) US (71) Applicant: VAYU WEB, INC. [US/US]; Suite 206, 309 Mamaroneck Avenue, White Plains, NY 10605 (US). (72) Inventor: LENZ, Frederick, P.; 183 Old Field Road, Old Field, NY 11733 (US). (74) Agents: MCNELIS, John, T. et al.; Fenwick & West LLP, Suite 700, Two Palo Alto Square, Palo Alto, CA 94306 (US).</p>	<p>(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).</p> <p>Published <i>Without international search report and to be republished upon receipt of that report.</i></p>	

(54) Title: SYSTEM AND METHOD FOR DISPLAYING INFORMATION AND MONITORING COMMUNICATIONS OVER THE INTERNET

(57) Abstract

A system and method for creating, transmitting, and receiving web pages from a web server. The present invention enables a user to create a new web site by either creating new web pages or extracting information from previously created web pages. The invention enables a web site master to control the entire presentation of data to the user. In one embodiment of the present invention, a web site can include a first web page having a main menu. The present invention then permits the web site master to prepare a preset sequence of web pages that are displayed to the user for each menu item. The user of the web page selects one menu item from a menu display and then can passively receive information in a logical sequence. The control of the web page sequence can be used by advertisers, for example, to effectively present all information deemed helpful in attempting to convince the user to use the advertised product or service. In contrast to conventional web sites, a user accessing a web site developed according to the present invention will have a predefined sequence of web pages displayed in order to maximize the benefit to the web site master, e.g., to increase the behavioral modifications of the client due to controlled advertising. The present invention also increases the effectiveness of web sites having multiple web pages by using an artificial intelligence preloading methodology to reduce the delay in displaying a web page on the user's computer.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon	KR	Republic of Korea	PL	Poland		
CN	China	KZ	Kazakhstan	PT	Portugal		
CU	Cuba	LC	Saint Lucia	RO	Romania		
CZ	Czech Republic	LI	Liechtenstein	RU	Russian Federation		
DE	Germany	LK	Sri Lanka	SD	Sudan		
DK	Denmark	LR	Liberia	SE	Sweden		
EE	Estonia			SG	Singapore		

18 minutes to download by the typical home computer, and only if the server that is sending the data is transmitting the data at the maximum permitted rate. While the data is being downloaded, conventional web sites display partial information on the screen of the users display monitor.

5 One growing use of the Internet and WWW is the creation of web sites for advertising. Advertisers are drawn by the low cost of the WWW. However, a significant drawback for advertisers is the slow rate of data transfer to typical home users. If an advertiser creates a web site where one page of the web site has a large size, e.g., two megabytes, then, as described above, a typical user having a 14.4 kbps modem will require over 18 minutes to download the
10 single web page. A significant number of users will be reluctant to download such large advertisement web pages due to the large time commitment.

Another significant drawback for advertisers is that conventional web sites require the user to search hypertext and hypergraphics in each web page in order to proceed to what the user thinks is of interest. Therefore, an advertiser operating a web site having several pages
15 frequently has users browse the web pages in a haphazard manner. However, successful advertisers prefer more control of the presentation sequence of an advertisement.

Accordingly, what is needed is a system and method for operating a web site that: (1) decreases the complexity of current web sites to reduce the intimidation of less sophisticated users of the web site; (2) decreases the waiting time between web page displays without
20 increasing the data transmission rate; and (3) enables a web site owner to have more control over the presentation of web pages to a user.

SUMMARY OF THE INVENTION

The present invention is a new model for creating, transmitting, and receiving web pages
25 from a web server. The present invention enables a user to create a new web site by either creating new web pages or extracting information from previously created web pages. The invention enables a web site master to control the entire presentation of data to the user. In one embodiment of the present invention, a web site can include a first web page having a main menu. The present invention then permits the web site master to prepare a preset sequence of
30 web pages that are displayed to the user for each menu item. The user of the web page selects one menu item from a menu display and then can passively receive information in a logical sequence. The control of the web page sequence can be used by advertisers, for example, to effectively present all information deemed helpful in attempting to convince the user to use the

advertised product or service. In contrast to conventional web sites, a user accessing a web site developed according to the present invention will have a predefined sequence of web pages displayed in order to maximize the benefit to the web site master, e.g., to increase the behavioral modifications of the client due to controlled advertising.

- 5 The present invention also increases the effectiveness of web sites having multiple web pages by using an artificial intelligence preloading methodology to reduce the delay in displaying a web page on the user's computer.

BRIEF DESCRIPTION OF THE DRAWINGS

- 10 Figure 1 is an illustration of a communication system in which the preferred of the present invention resides.

 Figure 2 is a more detailed illustration of the server of the preferred embodiment of the present invention.

- 15 Figure 3 is a more detailed illustration of the server storage device of the preferred embodiment of the present invention.

 Figure 4 is a flow chart describing the method performed by the server module according to one embodiment of the present invention.

 Figure 5 is a flow chart describing the step of analyzing a web site according to one embodiment of the present invention.

- 20 Figure 6 is an illustration of a computer screen display showing information extracted from the web pages of the web site during the analyze web site 408 process according to one embodiment of the present invention.

 Figure 7 is flow chart describing the step of configuring a modified web site according to one embodiment of the present invention.

- 25 Figure 8 is an illustration of a computer screen display for use in the web site configuration process according to one embodiment of the present invention.

 Figure 9 is an illustration of an edit transition display according to one embodiment of the present invention.

- 30 Figure 10 is an illustration of a configuration image menu display according to one embodiment of the present invention.

 Figure 11 is an illustration of a client computer storage device according to one embodiment of the present invention.

Figure 12 is a flow chart describing the method performed by the Vayu Web client module according to one embodiment of the present invention.

Figure 13 is a flow chart describing the method performed by the web page preload module according to one embodiment of the present invention.

5 Figure 14 is an illustration of a web page control panel according to one embodiment of the present invention.

Figure 15 is an illustration of a first menu web page according to an alternate embodiment of the present invention.

10 Figure 16 is an illustration of a second web site menu page according to an alternate embodiment of the present invention.

Figure 17 is an illustration of a site repository information according to an alternate embodiment of the present invention.

15 Figure 18 is an illustration of a screen display of the web site presentation properties of various web site sequences developed by the web site designer according to an alternate embodiment of the present invention.

Figure 19 is an illustration of a web page sequence properties screen display according to an alternate embodiment of the present invention.

Figure 20 is an illustration of a web page property screen display according to an alternate embodiment of the present invention.

20 Figure 21 is an illustration of a sequence optimizer screen display according to an alternate embodiment of the present invention.

Figure 22 is an illustration of the site layout screen display that is displayed to the web site designer according to an alternate embodiment of the present invention.

25 DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A preferred embodiment of the present invention is now described with reference to the figures where like reference numbers indicate identical or functionally similar elements. Also in the figures, the left most digit(s) of each reference number correspond(s) to the figure in which the reference number is first used.

30 Figure 1 is an illustration of a communication system in which the preferred embodiment of the present invention resides. The communication system includes a web server 102, a wide area network 104, destination servers 106A, 106B, and personal computers 108A-C. The personal computers 108A-C are conventional personal computers, e.g., an Apple computer

or a Compaq computer. The personal computers 108 can be independent or part of a network, for example. The personal computers 108 can be connected directly to the WAN, using a modem, for example, or can be connected to the WAN 102 via a server 106A that uses a leased line, for example. The server 106A can operate using a conventional operating system, for example, UNIX or Windows NT. In order for a user of a computer, e.g., destination computer 108A to access data representing world wide web pages (web pages) from a web server 102, the user of the personal computer 108A identifies a uniform resource locator (URL) address to a web page in the web server 102 and transmits a request signal to the WAN 104. The WAN 104 is conventional, e.g., the Internet, and can use routers to route the request signal to the web server 102. A conventional web server receives a request and transmits the requested web page to the destination computer 108A. The destination computer 108A receives the web page data and displays the web page. This process can repeat several times for each web page requested by the user. The multiple requests per page are to obtain each multimedia object embedded in the page, e.g., pictures, sounds, and videos.

The present invention is a new model for creating, transmitting, and receiving web pages from a web server 102. The invention includes a server module and a client module. The server module is stored and implemented by the web server 102. The client module is stored in the web server 102 and is transmitted from the web server 102 to the destination computer 108A after the web server 102 receives a request from the destination computer 108A. The client module is executed by the destination computer 108A when displaying the web pages of the web site. A more detailed discussion of the server module and the client module is set forth below.

Figure 2 is a more detailed illustration of the web server 102 of the preferred embodiment of the present invention. As stated above, the web server is conventional and includes a conventional central processing unit 202, e.g., a Pentium Pro processor that is commercially available from Intel Corporation, Santa Clara, California, a conventional input/output system 204, conventional random access memory 206, and a conventional storage device 208 having unconventional computer application programs stored therein. The computer application programs stored in the storage device 208 are described in detail below.

Figure 3 is a more detailed illustration of the web server storage device 208 of the preferred embodiment of the present invention. The storage device 208 includes a server web site module 302, a web site project database 304 having menu information 308 and web page information 306, a web site analyzer 310, and a web site configuration module 312. The server

web site module 302 includes web site data including MIME files. MIME files can include MIME types that provide a technique for enabling web browsers, e.g., Netscape Navigator, to automatically launch a viewing program associated with a received file, hypertext markup language (HTML) data, and common gateway interface (CGI) data. The web site project databases 304 includes menu information 308 and data from web pages 306 extracted from the web page by the web site analyzer 310. A more detailed description of the web site project database is set forth below. The web site analyzer 310 analyzes each web page of the web site, extracts data from each web page, and stores the data in the web site project database 304. The web site configuration module 312 enables a user to create a menu and to create an ordered sequence of web pages that can be presented to a user (client) when a menu item is selected.

Figure 4 is a flow chart describing the method performed by the server module 102 according to one embodiment of the present invention. If the desired web pages of a web site do not yet exist 402 then the web pages are created 404 using conventional web page creation techniques. A web site creation description is provided in Chandler, Running a Perfect Web Site, Que Corporation (1995) that is incorporated by reference herein in its entirety. If the web pages that will be part of the web site exist 402 then the server 102 defines 406 a new site project, analyzes 408 the web pages of the web site, configures 410 the new web site, and saves 412 the web site. A more detailed description of steps 406, 408, 410, and 412 is set forth below. Alternatively, web pages for the web site can be developed concurrently using the present invention.

As stated above, the web server 102 defines 406 a new site project. In one embodiment a new site project is a data structure that is stored in the server storage module 208. When fully defined, the new site project includes or references all of the information about a web site. The information stored about each web site includes general information about the web site and a list of the web pages in the web site. The general information includes information such as: (a) the company's name where the company can be the party to which the web site information is directed, e.g., the company who is advertising on the web site; (b) the webmaster name, e.g., the creator of the web site; (c) the webmaster electronic-mail (e-mail) address. The list of web pages includes information about each web page, such as: (1) the URL of the web page; (2) a list of links out of the page; (3) a list of links into the page from other web pages in the web site; (4) the title of the web page; (5) the content type of the page, e.g., text, html, and image; (6) a list of pictures/videos/sounds in the page; and (7) the size of the web page. The size (in bytes) of the web page can be determined using any of a variety of techniques. In one embodiment, the size

of the web page is the sum of the number of bytes in the page's HTML, the size of all image , sound <BGSOUND>, object <OBJECT>, applet <APPLET>, and script <SCRIPT> tag files. The information stored about each web page is determined during the steps of analyzing 408 the web site and configuring 410 the web site as described below.

5 Figure 5 is a flow chart describing the step of analyzing 408 a web site by the web site analyzer 310 according to one embodiment of the present invention. The web site analyzer determines the size of the web page and determines the links into and out of the page as described below. The web site analyzer 310 determines 502 if more web pages need to be analyzed. The web site analyzer 310 selects a web page by selecting one of the hyperlinks of a
10 previously analyzed web page. The web page analyzer 310 analyzes the web page and sets a flag indicating that the web page has been analyzed. If the web site analyzer 310 determines that the web page has already been analyzed, then the web site analyzer sets a pointer to the memory location in the server storage 208 containing the information previously extracted during the web page analysis.

15 The web page analyzer 310 extracts 504 the title of the web page. In one embodiment, the title of the web page is extracted by identifying the text associated with the title tag (<TITLE>). The title can be displayed in order to identify the web page to the user. The web page analyzer 310 then extracts 506 the hyperlinks of the web page. In one embodiment, the hypertext links are extracted by identifying the anchor tags (<A>), the frame tags (<FRAME>),
20 and map tags (<MAP>) in the web page. The web page analyzer 310 extracts 510 image file names by, for example, identifying the file associated with the image tag (), and the web page analyzer 310 determines the size of the image file by (a) reading the HTTP size header or (b) downloading the file. In one embodiment, the HTTP size header is read since it is typically faster than downloading the file. The web page analyzer 310 also extracts 512 embedded file
25 names in the web page by, for example, identifying the file associated with the embedded tag (<EMBED>). Then the web page analyzer 310 determines the size of each embedded file by (a) reading the HTTP size header or (b) downloading the file. The web page analyzer 310 also extracts 512 object files in the web page by, for example, identifying the file associated with the object tag (<OBJECT>). Then the web page analyzer 310 determines the size of each object file
30 by (a) reading the HTTP size header or (b) downloading the file. The web page analyzer 310 also extracts 512 sound files in the web page by, for example, identifying the file associated with the bgsound tag (<BGSOUND>). Then the web page analyzer 310 determines the size of each sound file by (a) reading the HTTP size header; or (b) downloading the file. The web page

analyzer 310 extracts 514 class file names by, for example, identifying the file names associated with a class file name tag, e.g., <APPLET> in the Java programming language commercially available from Sun Microsystems, Inc., Santa Clara, California. The web page analyzer then determines a list of links out of the web page by downloading and scanning the contents of each class file. The web page analyzer 310 also extracts 516 additional information from the web page. For example, the web page analyzer 310 identifies scripts, e.g., Javascripts and VBScripts, associated with a script tag (<SCRIPT>), downloads the scripts and scans the scripts for links out of the web page. After extracting the above identified information, the web page analyzer 310 determines 518 the size of the web page by summing the number of bytes in (1) the HTML of the web page, and (2) all of the image/embed/class/bgsound/object/script tag files. Steps 504-518 are repeated for each web page.

Figure 6 is an illustration of a computer screen display showing information extracted from the web pages of the web site during the analyze web site process 408, described above, according to one embodiment of the present invention. The computer display illustrated in Figure 6 displays web page information including the web site name 602, "http://McJava/samplertree/morelink.htm", and the status of the analysis 612, a current page identifier 614, the total number of pages 616, the total number of links found in all of the analyzed pages 618. In addition, the display includes information concerning the parsed pages 620 and parsing details 622. The parsed pages 620 shows the list of web pages as they are being parsed. The pages are displayed in a tree structure format that shows the hierarchical nature of the web site. The parsing details 622 shows a detailed list of the links and files found in the page being currently parsed. The site name 602 is the URL of the web site being analyzed. Host information (not shown) can also be displayed. The host is the name of the host machine on which the web site resides. Similarly, a tree level indicator (not shown) can also be displayed. The tree level represents the hierarchical level of the web page in the web site. Any web page may exist at a number of levels (e.g., it may appear as a hyperlink in many different pages in the web site). The level represented here is the level closest to the top of the tree that the page appears. It will be apparent to persons skilled in the art that more or less information can be available to the webmaster when configuring the web site.

After the web site analyzer 310 analyzes 408 the web site, the web site configuration module 312 configures the web site. Although the web site analyzer 310 has searched and analyzed each web page in the web site, the web site analyzer 310 does not modify the original web site. The web site configuration module 312 enables the web site developer, e.g., a

webmaster. to construct a menu. In one embodiment of the present invention this menu is typically approximately 8-10 items, although it is envisioned that any number of items can be part of the menu. Each item will consist of a sequence of web pages that are displayed in a fixed order. In one embodiment of the present invention the web pages themselves are not
5 modified except for the web page having the menu. In other embodiments the web pages are modified to, for example, ensure that all links out of the web page are deactivated, except for the links defined by the user. Figure 7 is a flow chart describing the step of configuring a modified web site by the web site configuration module 312 according to one embodiment of the present invention. The web site configuration module 312 enables the webmaster to create a menu
10 structure that can be the first web page displayed in the web site. For example, Figure 8 is an illustration of a computer screen display for use in the web site configuration process according to one embodiment of the present invention. The display includes a menu configuration portion 802, a current web site structure portion 804, a status portion 814, and various control buttons that can be selected by the webmaster using a conventional mouse or other pointing device. For
15 example. The control buttons in one embodiment include a menu shuffle up button 806, a menu shuffle down button 808, an add image button 810, an add transition button 812, and a status display 814.

The webmaster determines 704 if any more menu selections need to be modified. If the webmaster decides to modify or define a menu selection, the webmaster selects a menu, and
20 defines 706 a sequence of web pages that will be displayed when the menu is selected. One process for assigning a web page to a menu is: (a) a user highlights the menu in the menu configuration portion 802 using the mouse or keyboard; (b) the user then highlights the web page in current web site structure portion 804; and (c) the user selects the Add button 816. The chosen web page will be added to the menu and will be displayed in the menu configuration
25 portion 802 of the display. For example, a first menu is identified as "Luxury Vehicles" 820 in the menu configuration portion 802 of the display. The menu configuration portion 802 also includes information about the web pages that the webmaster has associated with the menu. For example, the top menu item is "Luxury Vehicles" 820. There are two web pages 822/824 associated with this menu item. These two web pages will be displayed in this order, i.e., web
30 page 822 and then web page 824, when the user ultimately views this web site with their client PC and browser.

The shuffle up button 806 and the shuffle down button 808 enable a user to modify the web site structure by moving a web page up or down in the displayed menu configuration

portion 802 of the display. These buttons simplify the web site editing procedure for the web master. For example, a web page can be selected by the webmaster and associated with one of the menu items. The webmaster can then move, add, delete, or copy web pages in order to develop a suitable web site presentation sequence. The webmaster can also introduce transitions that can control the transition between web pages by selecting the "Add Transition" button 812. For example, a transition may be a timer of 10 seconds, or a keyclick from the user. In one embodiment the webmaster ensures that the user will view the web pages associated with each menu (if any) only in the predefined sequence by automatically showing the next page in response to the transition event. For example, if a timer of 10 seconds has been selected by the web master then after 10 seconds has elapsed the next page will automatically be displayed. The user does not need to perform any action. In alternate embodiments, a limited number of page links are included that enable the user to advance to the next page, return to the previous page, or return to the main menu, for example.

As noted above, conventional web sites are created such that a client who accesses the web site can view the various pages in the web site in almost any sequence. One potentially significant use of the Internet and the WWW is in advertising. In one embodiment, the present invention can be utilized to create a web site having advertisement as a significant purpose of the web site. For example, an automobile company, e.g., General Motors, can develop a web site that advertises its currently available cars and trucks. One important aspect of advertising is the process of creating a need that can be satisfied by the advertiser. In order to efficiently and successfully accomplish this, it is helpful if the advertiser is able to control the sequence of information presented to the user. For example, if a car manufacturer developed a web site, they may not want the client to immediately jump to the web page having the cost of the vehicles. Instead, the advertiser could be more successful and effective if a description of the features and benefits of one or more car models were first shown to the user in order to familiarize the client with the cars and to therefore help persuade the client that the cars offered by the manufacturer will satisfy the needs of the client. After providing the information to the user, the web page having pricing information can be presented to the user. It will be apparent to persons skilled in the art that the above example is merely one example of how the current invention could be used and that many alternatives exist for developing and strategically planning the web page presentation sequence in a web site. The present invention provides the necessary tools to enable a webmaster to quickly, efficiently, and effectively generate a web site that gives control of the web site presentation to the web site owner instead of the client.

The page link can be displayed as a "Next Page" button or the webmaster can select to have one or more of the web pages transition automatically. Figure 9 is an illustration of an edit transition display 902 according to one embodiment of the present invention. In the example illustrated in Figure 9 the user has the ability to define the event that will trigger a web page transition for any particular web page, e.g., the "Leasing Information" web page as displayed in the menu item display area 904. The available types of transitions are displayed in the available transition portion 906 of the display. The webmaster selects one of the available transition types and activates the Add button 912, and the selection is displayed in the current transition portion 908 of the display. If the webmaster selects the timer option, the webmaster can set the delay between pages by entering the delay in the delay definition portion 910 of the display. To remove a selection from the Current Transition 908 display the web master uses the Remove button 914.

The webmaster also has the opportunity to provide 708 additional information concerning the web site and the individual web pages. For example, some of the information stored in the new site project data structure is defined by the webmaster, e.g., information concerning the company name, webmaster name, webmaster e-mail, and the URL of the main menu page or one or more web pages in the web site, for example. The webmaster has the opportunity to provide this information before or after the defining 706 the web page display sequence for the web site.

Figure 10 is an illustration of a configuration image menu dialog box display 1002 according to one embodiment of the present invention. The dialog box 1002 enables the user to generate a menu display by selecting a menu, e.g., Menu_4, and defining the active regions in the display portion 1006 for the selected menu. In Figure 10, the active region for Menu_4 ("New for '97") is defined in the active region definition dialog box 1008. In some embodiments of the present invention the webmaster does not create an initial menu page, instead only one web page sequence is defined and when the client selects the first page the sequence of the remaining web pages of the web site are displayed to the client in the predefined sequence.

After the webmaster defines the menus and defines the web page sequence associated with each menu, the modified web site is saved 412. In one embodiment of the present invention, the web site is saved as a series of files that are used when web site information is transferred to a client accessing the web site. In one embodiment these files are used to transfer the information gathered by the server tool to the client. The files include a stub Java Applet that is inserted into the home page of the site. One example of such a file is set forth in Table 1.

<APPLET CODE="browser.class" WIDTH="10" HEIGHT="10">

5 <PARAM NAME="menu_1" VALUE="[MENU_ITEM NAME='Luxury
Vehicles' AUTO=TRUE RECT=199,92,230,34 IMG=menu.gif]">

10 <PARAM NAME="menu_1_1" VALUE="[LINK
URL=http://www.xyzmotors.com/L2_N2.HTM SIZE=3514
TRANSIT=TIMER DELAY=10]">

15 <PARAM NAME="menu_1_1_1" VALUE="[LINK_OBJECT
URL=http://www.xyzmotors.com/Images/I_1_9.GIF
SIZE=1059 CONTENT=image/gif]">

20 <PARAM NAME="menu_1_2" VALUE="[LINK_FILE
NAME=xyzmotors_menu_1.brw]">

25 <PARAM NAME="menu_2" VALUE="[MENU_ITEM NAME='Sport Utility'
RECT=199,129,229,39 IMG=menu.gif]">

30 <PARAM NAME="menu_2_1" VALUE="[LINK URL=http://
www.xyzmotors.com/L1_N1.HTM SIZE=12117
TRANSIT=MENU]">

35 <PARAM NAME="menu_2_1_1" VALUE="[LINK_OBJECT
URL=http://www.xyzmotors.com/Images/I_1_1.GIF
SIZE=1257 CONTENT=image/gif]">

40 <PARAM NAME="menu_2_2" VALUE="[LINK_FILE NAME=
xyzmotors_menu_2.brw]">

45 <PARAM NAME="menu_3" VALUE="[MENU_ITEM NAME='Trucks and
Vans' RECT=198,173,230,47 IMG=menu.gif]">

50 <PARAM NAME="menu_3_1" VALUE="[LINK URL=http://
www.xyzmotors.com/L1_N1.HTM SIZE=12117 TRANSIT=TIMER
DELAY=10]">

55 <PARAM NAME="menu_3_1_1" VALUE="[LINK_OBJECT
URL=http://www.xyzmotors.com/Images/I_1_1.GIF
SIZE=1257 CONTENT=image/gif]">

<PARAM NAME="menu_3_2" VALUE="[LINK_FILE NAME=
xyzmotors_menu_3.brw]">

<PARAM NAME="menu_4" VALUE="[MENU_ITEM NAME='Family Fare'
RECT=195,224,232,40 IMG=menu.gif]">

<PARAM NAME="menu_4_1" VALUE="[LINK URL=http://
www.xyzmotors.com/L1_N1.HTM SIZE=12117 TRANSIT=TIMER
DELAY=10]">

<PARAM NAME="menu_4_1_1" VALUE="[LINK_OBJECT
URL=http://www.xyzmotors.com/Images/I_1_1.GIF
SIZE=1257 CONTENT=image/gif]">

<PARAM NAME="menu_4_2" VALUE="[LINK_FILE NAME=
xyzmotors_menu_4.brw]">

```
<PARAM NAME="menu_5" VALUE="{MENU_ITEM NAME='New For
'97'!"}>
```

```
5 <PARAM NAME="menu_5_1" VALUE="{LINK URL=http://
www.xyzmotors.com /L1_N1.HTM SIZE=12117 TRANSIT=TIMER
DELAY=10}">
```

```
10 <PARAM NAME="menu_5_1_1" VALUE="{LINK_OBJECT
URL=http:// www.xyzmotors.com /Images\I_1_1.GIF:
SIZE=1257 CONTENT=image/gif}">
```

```
15 <PARAM NAME="menu_5_2" VALUE="{LINK_FILE NAME=
xyzmotors _menu_5.brw}">
```

```
</APPLET>
```

Table 1

20 These parameters in Table 1 contain specific information about each link on the menu illustrated in the display portion 1006 of Figure 10, and the contents of the pages linked thereto. The menu items are passed as parameters. In the example set forth in Table 1, a naming convention is used to determine the order and structure of the menu. The data is structured with a format that is similar to HTML, but uses square brackets ([and]) instead of angle brackets (< and >). The first parameter (PARAM) is for the first item in the menu, named *menu_1*. Its type is *MENU_ITEM*, and the item's name is *Sport Utility*. The next *PARAM* is a page assigned to the *menu_1*, and its name is *menu_1_1*. The type is *LINK*, its *URL* is <http://McJava/dealers/dealers.html>, the *TRANSIT* is *TIMER*, and the *DELAY* is 10 seconds. The third *PARAM* is also assigned to the *menu_1* item, but it points to a file that contains the rest of the links for *menu_1*. The file name is *McJava_menu_2.brw*. There are one of these files for each menu item that has more than one link assigned to it. These files are used to limit the amount of information that is transmitted from the server to the client in the initial transmission. Since the number of web pages that could be assigned to all of the menu items could range into the hundreds and thousands, it is generally more efficient to send down a portion of menu information, and then retrieve more menu information as needed. The file structure design is extensible such that as new information is needed new name/value pairs are added. Older version of the software will ignore name/value pairs they do not understand. Newer version of the software will have reasonable defaults if they encounter an older file with missing name/value pairs.

40 As described above, a client utilizes a computer, e.g., a destination computer 108A, to access the defined web site stored on the server 102. The personal computer used by the client

can be a conventional personal computer, e.g., an IBM personal computer that is commercially available from IBM Corporation, Armonk, NY. The destination computer 108A includes a conventional CPU, conventional RAM, and a conventional storage device, that can be similar to the server 102 described above. Figure 11 is an illustration of a client computer storage device 5 1100 according to one embodiment of the present invention. The storage device 1100 includes a conventional Internet browser 1102, a Vayu Web client module 1104, a web page preload module 1106, menu files 1108, and a TCP/IP Internet connection protocol module 1110. The conventional Internet browser provides an Internet gateway to the destination computer 108A. The client uses the Internet browser 1102 to request access to the web site stored in the server 10 storage module 208. The menu files 1108 include the files received from the server 102 in response to the access request. In addition, the personal computer storage module 1100 receives the Vayu Web client module 1104 and the web page preload module 1106 from the server 102. The operation of these modules is described in greater detail below.

In one embodiment, the Vayu Web client module has been implemented using three 15 different techniques. These three techniques allow the client module to operate across the wide variety of client machines currently in use across the Internet, e.g., IBM PC, Apple Macintosh, Sun Microsystems Sparcstation, and across the wide variety of WWW Browsers currently in use, e.g., Netscape versions 2.n and 3.n, and Microsoft Internet Explorer versions 2.n and 3.n. The three techniques are: (1) as a Java Applet; (2) as a Javascript script where Javascript is a 20 Internet language developed by Netscape Corporation; (3) as a Client-Pull mechanism using a <META> tag in the HTML file, that causes the browser itself to automatically load pages on a timed basis. The CONTENT argument contains the time in seconds. The URL contains the target URL. The <META> tag can be:

```
<META HTTP-EQUIV=REFRESH CONTENT="1 URL=target_url">
```

Figure 12 is a flow chart describing the method performed by the Vayu Web client 25 module according to one embodiment of the present invention. As described above, the client requests information from the web site server 102 and the destination computer 108A downloads 1202 web site server information. The web site server information initially downloaded includes the Vayu Web client module 1104, the web page preload module 1106, 30 and the new site project data structure that is stored in the menu files 1108. As described above, the new site project data includes information about the web site and includes references to web pages in the web site but does not include all of the web page data. The destination computer 108A implements 1204 a web page preload routine that is executed concurrently with the

remaining functions performed by the Vayu Web client module, as described below. Figure 13 is a flow chart describing the method performed by the web page preload module 1106 according to one embodiment of the present invention. The web page preload module 1106 gathers local environmental information concerning the client personal computer. This information includes the speed of the modem, the time of day, Internet service provider service level, browser type and version, and operating system type and version. This information is used by the artificial intelligence expert system to dynamically select a preload strategy to optimize the performance of the presentation of web pages to the client. The web page preload module 1106 preloads 1304 the first web page from each of the menus.

While the web page preload module 1106 is preloading the first web page associated with each of the menus, the Vayu Web client module 1104 displays 1206 the menu page and waits for the user to select a menu. When the user selects 1208 a menu item, the Vayu Web client module 1104 identifies 1210 the first web page in the predefined sequence of web pages associated with the menu item and displays 1212 the first web page. Before the user selects a menu item, the web page preload module 1106 preloads the first web page associated with each menu item, as stated above, and then determines the web page loading sequence for subsequent web pages based upon a determination of the most likely menu item selection. The web page preload module 1106 determines the most likely menu item selection based upon the selections of previous clients or, if no clients have previously accessed the web page, the webmaster may indicate which menu item to show first, or the first menu item is selected as the most likely to be selected. The web page preload module 1106 preloads web pages from the sequence of web pages associated with the most likely menu item selection by implementing the rule based expert system of the present invention, described below.

After the client selects a menu item, the Vayu Web client module 1104 and the web page preload module 1106 receive the web page sequence and web page information from the new site data structure in the menu files 1108. Specifically, the new site data structure includes the overall size of the web page, the number of embedded objects in each web page, and the size of the embedded objects in each web page, as described above. In one embodiment of the present invention, the web page preload module 1106 determines a priority for each web page in the web page sequence based upon the environmental parameters of the destination computer 108A, described above, a queue location of the web page in the predefined sequence, the size of the web page, the number of embedded objects in the web page, and the size of the embedded objects in the web page, for example. In addition, the web page preload module is keeping track

of how long it actually takes to load each page. From this information it calculates the actual throughput that the current client is getting. This throughput is used by the expert system to make decisions about preload strategies. For example, if the throughput falls below a certain value, the preload module may stop preloading certain types of MIME files (e.g., video, sound).

5 The trigger value would be set by the web master during the menu configuration stage described above. One example of a rule based expert system is set forth in Table 2.

```
10 ; Set preload strategy based upon throughput
    IF
        system.throughput < 6kpbs
    THEN
        preload.order = STRATEGY_7
    ENDIF

15 ; Set preload strategy based upon throughput
    IF
        system.throughput < 2kpbs
    THEN
        mime.image = NOLOAD
    ENDIF

20 ; Set priority based upon size of an embedded object
    IF
        object1.size > 250000
    THEN
        object1.preload_priority = 5
    ENDIF

25 ; Set priority based upon size of an embedded object
    IF
        object1.size > 500000
    THEN
        object1.preload_priority = 7
    ENDIF

30 ; Set priority based upon size of an embedded object
    IF
        object1.size > 1000000
    THEN
        object1.preload_priority = 10
    ENDIF

35 ; Set priority based upon the number of embedded
    ; objects in a page
    IF
        page1.objects > 5
    THEN
        page1.preload_priority = 3
    ENDIF

40 ; Set priority based upon the number of embedded
    ; objects in a page
    IF
```

```

    pagel.objects > 10
  THEN
    pagel.preload_priority = 8
  ENDIF
5
  ; Set preload order based upon browser type and time
  of
  ; day
  ; MSIE_3x is Microsoft's Internet Explorer Version 3.x
10 IF
    ; night time hours
    browser.type = MSIE_3x AND (env.timeofday > 20:00
  AND env.timeofday < 08:00)
  THEN
15   preload.order = SEQUENTIAL
  ENDIF

  ; Set preload order based upon browser type and time
  of
20   ; day
  ; STRATEGY_7 is load page 1, then 2, then largest to
  ; smallest
  IF
25   ; business hours
    browser.type = MSIE_3x AND (env.timeofday >
  08:00 AND env.timeofday < 20:00)
  THEN
    preload.order = STRATEGY_7
30  ENDIF

```

Table 2

After determining the preloading priority for each web page, the web page preload module 1106 preloads 1308 the web pages having the highest priority. The web page preload module 1106 monitors 1310 the preloading procedure to reduce any delay in displaying web page to the client. The monitoring function of the web page preload module 1106 permits the preload function to be dynamic in that if a problem occurs 1312 in the web page loading process, the web page preload module 1106 sorts and reorganizes 1314 the data. Examples of problems include: (a) when a page is needed to be displayed and has not yet been preloaded or (b) when the preloading is sufficiently ahead of the display so that the local PC web page cache is full, and pages must be discarded (and reloaded). When a problem occurs the expert system is re-initialized with the current data (including the throughput and modified menu files without the web pages that have already been displayed). A modified preload strategy is then determined 1306.

45 The client continues navigating 1210 through web pages while the destination computer 108A continues displaying 1212 web pages. As described above, in one embodiment of the present invention, the client can passively watch the sequence of web page displays. In alternate

embodiments, the user has the option of selecting a next page button. Within each web page the client also has the option of manipulating a control panel. Figure 14 is an illustration of a web page control panel 1400 according to one embodiment of the present invention. In this embodiment, the user has the option of returning to the beginning of the web page by selecting control panel button 1402, proceeding to the end of the web page (for example, a video clip embedded in the web page may be skipped) by selecting control panel button 1404, continuing or beginning the execution of the web page display by selecting control panel button 1406, pausing the execution of the web page by selecting control panel button 1408, decreasing the speed of web page execution by selecting control panel button 1410, increasing the speed of web page execution by selecting control panel button 1412, or stopping web page execution by selecting control panel button 1414.

An alternate (preferred) embodiment of the present invention is described below with reference to Figures 15-22. This alternate embodiment includes a standard structure and navigation for the menus (pages containing buttons) and menu items (buttons), allowing the user a set of options at each web page. One benefit of using the standard structure is to make the web site easier to use by the end user and to offer the web site designer more control over what is displayed to the user.

Figure 15 is an illustration of a first menu web page according to an alternate embodiment of the present invention. The first menu page 1500 illustrated in Figure 15 includes a single large button 1502 that includes the name of the company that is the subject of the advertising, e.g., "Vayu Web". The user can select the button 1502 to continue or the user can exit from the web site. In addition, the present invention enables the company to add a graphic display to the button or to the background to enhance the screen display.

The first menu web page can include a tool bar 1504 having various control buttons thereon. These control buttons are enabled and disabled for each web page based upon the web site definitions defined by the webmaster. When the user selects the button 1502, a sequence of pages may automatically be presented after which another menu page appears containing more items. Alternatively, the sequence can be skipped and the next menu page can be presented immediately. While the user is viewing a dynamic sequence, the control functions are typically disabled. The control functions include pause, fast forward, reverse, and exit. In alternate embodiments these control functions are enabled during the presentation of dynamic sequences. While a dynamic sequence is playing the user typically watches the display until the next menu page is reached.

Figure 16 is an illustration of a second web site menu page according to an alternate embodiment of the present invention. At the next web site menu page 1600, the user has three options, (1) return to the previous menu page, (2) select one of the menu items 1602, or (3) select the auto-play button 1604 to automatically cycle through the sequences following each menu item, for example. The order of cycling through the menu items is determined by the web site designer (web site master). Some items can be excluded from this "autoplay". A menu page can comprise any number of items. In this alternate embodiment the number of items is typically between 1 and 10. The web site designer can add their own graphics to the items or to the page background.

10 Certain web pages can be static, i.e., they do not automatically display another web page. Frequently, for static web pages, the "pause" and "forward" buttons on the tool bar 1504 are enabled. In this case, the page may include a description of a product and/or ordering information, for example. A maximum viewing time can be specified by the web site designer to proceed to the next page after some amount of time has elapsed, or the web site designer can permit the user to continue to the next web page by selecting a next page control button.

In this alternate embodiment, the tool bar 1504 is located at the top of the screen. The web site designer may choose not to display the tool bar 1504 for certain screens when, for example, the user has no other options within these screens but to wait for the next page. The tool bar 1504 can also include status indicators 1606 to show the remaining time left to view this page. This tool bar 1504 can be used for every page displayed using the present invention.

Another feature of the present invention is that the user is permitted to create multiple versions of each web site where the present invention permits optimization of each web site for different internet throughput rates. For example, the throughput range for different types of Internet access devices commonly used today varies from 14.4 kbps (modem) to 56 kbps (ISDN) to 1.5 megabits/sec (T1). This variation exceeds a factor of 100. Furthermore, users of the present invention on Intranets, e.g., local area networks (LANs) can reach data transfer rates exceeding 10 megabits/sec. Due to transfer rate constraints at the lower throughput speeds, the web site designer may include several versions of a web page, or several versions of a sequence, where a sequence includes a series of pages which are automatically displayed, with each version being a different size to optimize the web site display for different throughput environments. The database can include information about the various page and sequence versions. The artificial intelligence module of the present invention that was described above with respect to the expert system and is described in greater detail below, can automatically

select the appropriate page version depending on the throughput environment it has determined for a user. The Virtual Design Studio, which is the tool that the web site designer uses to assemble the web pages into advertisement sequences, is designed to assist the web designer in creating optimal sequences for various throughput environments. The Virtual Design Studio will also allow the web site designers to preview the web pages of the web site using a simulator of the present invention in order to simulate environments having various data throughput rates.

It will be apparent to persons skilled in the art that the use of different versions of the web pages is not required to operate the present invention. However, if the web site designer is expecting users with different throughput environments to access the web site, it may be advantageous to create versions of certain web pages that contain videos, sound, etc., so users with faster connections can view web pages having greater detail, while users with lower connections can still get the full benefits of this web site by viewing pages of a smaller size.

The present invention permits different versions of web pages and web sites by providing the web site designer with the option of providing high-quality (larger) pages for users connected at higher throughput rates and lower-quality (smaller) pages for users connected at lower throughput rates. The present invention enables the web site designer to easily mix the smaller pages with the larger pages when the throughput is fast. However, adding larger pages to a low-throughput connection requires the designer to carefully sequence the pages so a large page is preloaded while several smaller ones are being viewed. In addition, some pages take longer to view, for example if they contain detailed textual information that takes longer to read. This longer viewing time can be used to preload larger files. The Virtual Design Studio of the present invention includes modules which assist in the intelligent sequencing of pages in this manner not only to reduce the user's waiting time but also to improve the quality of the pages that can be viewed with slower connections.

Another factor to consider is that the actual throughput rate during an Internet connection session can vary greatly. For example, with a nominal connection speed of 28.8 kbps, the actual transfer rate can easily vary between 10 kbps and 28.8 kbps. Therefore, the artificial intelligence (AI) module of the present invention makes real-time adjustments during a connection session to keep the presentation flowing and prevent waiting time for the user. One of the techniques implemented by the AI module is to leave out certain pages that the designer has identified as being large but with low impact or information content. Another technique is to dynamically switch a higher-quality presentation to a lower one when the throughput slows

down, and then to switch back to the higher-quality one if the throughput picks up again. The artificial intelligence unit is described in greater detail below.

Figures 6, 8, 9, and 10 are illustrations of screen displays during the analyze web site process 408 and the configure web site process 410 for one embodiment of the present invention as described above. Figures 17-22 are illustrations of screen displays displayed to the web site designer during the analyze web site process 408 and the configure web site process 410 according to an alternate embodiment of the present invention. The Virtual Design Studio is stored in the web site analyzer 310 and enables a web site designer to: (1) construct presentation sequences from pre-existing and newly created web pages; (2) optimize the presentation sequences for different throughput environments and specify alternatives to be used by the artificial intelligence module to compensate for throughput variations in real time; and (3) create menu structures and user navigation through the web site. The virtual design studio includes a site repository information screen that is illustrated in Figure 17. The site repository information screen allows a web site designer to view, insert, delete and edit web page information and web page sequences. For example, the site repository information screen allows the web site designer to assemble pages into the sequences which form the basis of the automated web site content. The site repository information screen also enables the web site designer to modify and maintain the web site.

Figure 18 is an illustration of a screen display of the web site presentation properties of various web site sequences developed by the web site designer. Figure 18 illustrates two alternate web sites, the first web site 1802 is defined to be accessed by clients having a high data transmission connection, e.g., a client having a T1 line. The second web site 1804 is defined to be access by clients having a low data transmission connection, e.g., a client having a 14.4 or a 28.8 kbps modem connection.

Figure 19 is an illustration of a web page sequence properties screen display 1900. The web page sequence properties screen display 1900 identifies the optimal data connection rate for a particular web site sequence. In Figure 19, the optimal user has a connection rate between 14.4 kbps and 33.3 kbps. The web page sequence properties screen display also enables the user to identify a slower alternate web page sequence and a faster alternate web page sequence that can be selected by the artificial intelligence unit, as described below.

Figure 20 is an illustration of a web page property screen display according to an embodiment of the present invention. The web page property screen display 2000 includes information about various components of each web page including the size and download time

of each component, whether the web page is static or dynamic, as defined above, and defines various timing features of the web page including the nominal load time, the ideal viewing time, the minimum viewing time and the maximum viewing time. In addition the page property enables the user to select various options including the display of control buttons, e.g., the forward and pause button, and enables the user to identify web pages that can be skipped if the actual data throughput is slower than the anticipated data throughput, as determined by the artificial intelligence unit, described below.

Figure 21 is an illustration of a sequence optimizer screen display according to an alternate embodiment of the present invention. The sequence optimizer screen display allows the user to optimize the arrangement of web pages within a web site and to optimize the length of time each web page is displayed. The sequence optimizer screen display is an easy-to-use graphical interface that displays the ideal and estimated viewing times for each page. The sequence optimizer is part of the web site configuration module 312 illustrated in Figure 3. The sequence optimizer of the present invention calls a web page display simulator that is also part of the web site configuration module 312 which provides feedback to the web site designer regarding the presentation of the sequence of web pages to the user under different network throughput conditions. The present invention determines the ideal and estimated viewing times based upon the amount of text in the web page, the level of detail in visual objects, and the length of video objects, for example. The user can modify the viewing time for each page.

Using this ideal viewing time and the estimated page loading time under various network throughput conditions, the Sequence Optimizer graphically displays for each page the amount of credit time, i.e., the time differential between the viewing time and the time at which the page has been preloaded, or the amount of debit time, the time differential between the time at which the page will be preloaded and the viewing time. The user can adjust the parameters for each page and see this credit/debit value change for each page. The parameters that can be set for each page include: the ideal view time, which is the amount of time the designer wants page to display, and the optional view flag, which indicates whether a page can be skipped over if the system is too busy, for example.

In addition the Virtual Design Studio unit automatically computes for each page the estimated minimum view time, which is the minimum time estimate that the page must be displayed before the next page is ready for display, and the normal load time, which is an estimate of the amount of time it will take to download a page in the throughput environment that the designer has set for this sequence.

The site layout studio unit is part of the web site configuration module 312 and enables the web site designer to create menus and define the web page sequence of the web site. The site layout studio unit displays a site menu page display and a menu item that displays the name of the site. When a user selects the menu item, one of at least two things happen: (1) a sequence of pages will play, or (2) another menu will appear, each of which can contain up to ten menu items. Using this model, sequences and menus can be nested infinitely. Figure 22 is an illustration of the site layout screen display that is displayed to the web site designer according to an alternate embodiment of the present invention. The site layout screen display 2200 includes a menu portion 2202, a menu layout portion 2204, and a control portion 2206. The menu portion 2202 includes the tree of menus and menu items within a site and the links there between. Selecting an object expands it and shows the menus and items linked hierarchically below it. The menu layout portion 2204 displays the screen layout of the menu items on each menu. Selecting a menu in the upper screen causes that menu to be displayed in the menu layout portion 2204. The user has the option of changing the menu style, by selecting the menu style button 2208 in the control portion 2206 in terms of the types of standard menu styles supported by the present invention, e.g., by modifying the displayed graphics and colors.

The virtual design studio unit includes a relational database that resides on the same computer the Virtual Design Studio is on, which can be the web site designer's computer. This database contains all the information regarding the automated web site except for the pages themselves, which are stored in files on the web server. A conventional HTML Generator module of the Virtual Design Studio creates an encoding of the database contents into a format that can be easily read by the client-side software on the user's computer. When the web site designer has completed defining the menus, pages, and sequences for the automated web site, the web site designer runs the HTML generator to create a file which is placed on the web server computer. These generated files contain the HTML code which the client-side processor and AI processing module uses to perform.

The web page preload module 1106 is described above with reference to Figures 11-13. In an alternate embodiment the web page preload module includes an artificial intelligence (AI) module that performs the functions described below. The artificial intelligence module supports the selection of different page versions and additional real-time decisions on page preloading. In other embodiments the artificial intelligence module can be stored on the server instead of the client computer.

After a user connects to the server having the web site, the AI module is downloaded to the client computer and collects information about the client computer environment including the operating system, processor capacity, Java support capabilities, types of video and audio formats that are supported, and the current throughput rate of the Internet connection. Using
5 this information the AI module makes decisions regarding which version of Java to use and which version of the page sequence to use. Since network conditions can change during the session, the AI module continually monitors the network conditions, e.g., the rate of data throughput, and can modify the web page sequence by skipping a web page, for example, in order to reduce the waiting time for the user. The AI module periodically assesses the state of
10 the session, and makes decisions accordingly. The functions performed by the AI module includes: (1) preloading pages; (2) canceling the preloading process; (3) modifying the rate of play of the current sequence; (4) inserting an additional page, e.g., a message, into the sequence; (5) skipping a page in a sequence being presented; (6) replaying one of the components, e.g., sound or video; and (7) switching to presenting another sequence of pages with a different ideal
15 throughput requirement. A more detailed description of the AI module operation is described below.

In the alternate embodiment the AI module is downloaded to the client computer. Specifically, when the user contacts the web site, a client module is downloaded on the first page and the HTML code output by the HTML generator module of the virtual design studio,
20 described above is also downloaded. This HTML code is used by the client code module and the AI module, and contains the complete description of the web site in terms of the menus, menu items, sequences, and pages found in each sequence. The client module performs three major functions: (1) probing the client machine to determine the client environment and the throughput rate, and uploading the HTML code; (2) displaying menus and pages; and (3)
25 processing user toolbar input.

The client module implements the site template as designed by the web designer. The first menu page is displayed, and the user can choose one of the available options. When the user selects an option, the sequence of web pages associated with that option is displayed in sequence on a display device, e.g., a computer monitor, coupled to the client computer. Each
30 web page can be displayed for a certain period of time, and the client module is responsible for switching pages after the view time has expired.

As stated above, some web pages may have an active toolbar. For example, some pages with a great deal of text may require that the user indicate when the user has finished reading the

text. It is possible to have a single view time that would work for all users. however fast readers may become frustrated if the view time is too long since the view time would need to be long enough to accommodate slower readers.

The client-server interaction and processing on the Internet World Wide Web is
5 complex, and is greatly affected by factors such as network traffic, the number of clients accessing a particular site, the amount of content the client requested, the computer hardware speed, and the software efficiency. While some of these factors (e.g. modem speed) stay the same during an Internet session, other ones, such as network traffic, varies constantly.

To deliver an uninterrupted or nearly uninterrupted presentation to the user, the AI
10 module frequently monitors the network, the server, and the client environment during a session, and makes complex decisions regarding how to proceed with the presentation. The AI module accounts for processes happening both on the server and on the client computer. The AI module synchronizes these processes while gathering information and while making decisions. The AI module constantly performs "what-if" scenarios, examining different options for
15 preventing interruptions to the pages appearing on the user's display screen.

The AI module relies on the sequences being constructed in an optimal manner for nominal throughput for a particular environment. For example, if the nominal throughput for the sequence is 20 kbps, the AI module presumes that the sequence was constructed to flow smoothly as long as the connection uniformly maintains this throughput. However, the present
20 invention is designed to work under various network conditions, under different server responsiveness, and with a variety of client computer environments, e.g., low-end personal computers with basic web browsers or high-end workstations supporting more powerful web browsers.

As described above, at the beginning of the client session, the AI module determines the
25 client's environment and the current state of the Internet connection. Some of the parameters which will affect the decisions of the AI module are: (1) the processing and Input/Output (I/O) speed of the server machine; (2) the throughput capabilities of the Web Server software; (3) the operating system on the client machine; (4) the client computer's support of Java applets; (5) the client computer's support of JavaScript; (6) the client computer's support of Client Pull
30 technology; (7) the types of video and audio formats that can be played by the browser or plug-in components of the client computer; (8) the maximum number of simultaneous connections to the Server that the client permits; (9) the browser cache size of the client computer; (10) the

number of users currently communicating with the Web Server; and (11) the current throughput rate of the Internet connection.

In one embodiment of the present invention, the AI module determines the following parameters at the beginning of a session and monitors the following parameters throughout the session: (1) the number of users currently communicating with the Web Server; (2) the current
5 throughput rate of the Internet connection; and (3) the volatility of the network throughput rate.

At the beginning of each presentation, the AI module, based on the parameters of the session: (1) selects the versions of the requested presentation that are supported by the client environment. e.g., Java, JavaScript, ClientPull, or Generic; and (2) selects the starting version of
10 the presentation. As described above, since network conditions may change during the presentation, the present invention can utilize all of the selected versions by, for example, switching between web page sequence versions in response to a variations in network conditions. In addition, the present invention sets the following parameters: (1) the maximum number of simultaneous connections the present invention utilizes; (2) the rate of change of the
15 speed of the presentation sequence in response to change in network conditions; and (3) the types of filler pages, filler applications and special effects to be used during the presentation. In addition, as the session progresses, the AI module preloads new pages, graphics, sound, video and other presentation components. The AI module uses the information obtained during its network monitoring operation to determine whether the original sequence still can be played
20 without interruptions. The AI module also monitors the progress of each component being preloaded from the Web Server, in order to make decisions about when to cancel preloading, or initiate new component preloading, as described above.

Appendix A includes a pseudo-code representation used to implement the AI module and is incorporated by reference herein.

25 The following is an example of the operation of the AI module. After the client connects to a server, the present invention determines, for example, that the client computer utilizes the Windows 95 Operating System, supports Java applets, supports running QuickTime videos, permits a maximum of two simultaneous connections to the server, and has a browser cache size of 10 megabytes. The AI module determines that versions of the top-level presentation can
30 utilize Java applets and QuickTime video technology.

A probe page is downloaded to the client computer, and the present invention determines that the number of users currently communicating with the web server is at about half of Server's capacity and the current throughput rate of this Internet connection is 1000 kbps.

The present invention then selects the starting version of the presentation that corresponds to network throughput rate of 1000 kbps, and preloads the first page of the web site. The AI module determines that there are two components currently being downloaded, one from the next page, and one from the following page.

5 The presentation begins, and the present invention starts presenting selected sequence of pages, while continuing to monitor the presentation and network conditions.

During the presentation, and as a result of the network monitoring performed by the AI module, the present invention detects a drop in network connection's throughput, for example down to 100 kbps.

10 The AI module of the present invention sets up a virtual scenario of canceling the process of preloading the second component because even though the first component would download sooner, the timing for the next page would still be late.

The AI module determines that current page has an animated GIF component that can be replayed at the end of the page and adds this replay to the scenario being considered. However,
15 invoking a what-if analysis still results in unacceptable timing. Therefore, the expert system consults a presentation script, and determines that at this point a "Skip page", or "Special Effect" actions are not available and, accordingly, rejects this scenario.

The expert system then runs through some other viable "what-if" scenarios, and rejects
20 all of them except the one requiring switching to the sequence corresponding to a 56 kbps throughput.

The AI module then instructs the Vayu Web client module 1104 to modify the selected sequence of pages. The AI module then continues monitoring the presentation and network conditions.

25 Later in the presentation, the AI module detects an increase in network connection's throughput up to 900 kbps. The expert system sets up a virtual scenario of switching to the sequence corresponding to 1000 kbps throughput, and runs a what-if analysis on that scenario, whose result is acceptable. The AI module then switches back to the sequence of pages corresponding to 1000 kbps throughput, and continues with the presentation.

30 While the present invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by persons skilled in the relevant art that various changes in form and details can be made therein without departing from the spirit and scope of the invention.

Appendix A

5 The following is Java-based pseudo-code for the AI module
for an embodiment of the present invention.

Class Definitions

```

10 class PrSite
{
    //describes the site (e.g. "General Goods")
    //for which presentations are built
    string name;
    PrSiteViewVec & siteViewVec; //presentation projects
15 };

class PrSiteView
{
20 //hierarchical tree of presentations accessible to user logging-in.
    string name;
    PrSite & site;
    PrPage & rootMenuPage; //top-level menu
    Presentation & rootPresentation; //top-level presentation
25 };

typedef Vector<Presentation> PresVec

30 class Presentation
{
    //describes a presentation corresponding to a menu item
    string name;
    string type; //sequence, white-paper, order-form
35 PrSiteView & siteView;
    Presentation & parent; //parent in the pres. tree
    PresVec & childVec; //first child in the pres. tree
    SequenceVec & sequenceVec; //list of alternate page-sequences
    PrPage & tailMenuPage; //menu to display after presentation
40 PrPageVec & tailPageVec; //list of pages tacked-on (e.g. order-
    //form)
};

45 class PrSequence
{
    //sequence of pages making up a presentation
    //a presentation may have several alternative sequences
    //for different communication speed, Java support, plug-in support
    string name;
50 string type;
    string environmentType; //e.g. for Java-supporting browsers
    Presentation & presentation; //back pointer to presentation owning
    PrSeqItemVec & itemVec; //list of items (pages) comprising this
55 };

class PrSequenceItem
{
    //item of the presentation sequence
    PrSequence & sequence;
60 PrPage & page; //page (usually HTML)
    long idealViewTime; //prescribed under normal state
    long minViewTime;
    long maxViewTime;
65 long loadTime; //expected load time at throughput
    //for which sequence was designed

```

```

    int         pauseAllowed; // user pauses when this item is displayed
    int         forwardAllowed; //enable "forward" button when displayed
    int         skipAllowed; //allow skipping this page
5   PrPage &   preloadPage; //when displaying this item, start preloading
                        //preloadPage before preloading next page
    PrFxVec &   fxVec;      //list of available "special effects"
                        //to be used when timing is Late
    PrSequence & upSequence; //another sequence to switch to
                        //when Internet conditions improve
10  PrSequenceItem & upItem; //item within sequence to switch to
                        //when Internet conditions improve
    PrSequence & downSequence; //another sequence to switch to
                        //when Internet conditions degrade
15  PrSequenceItem & downItem; //item within sequence to switch to
                        //when Internet conditions degrade
};

class PrPage
20 {
    //HTML page, or special effect page, or applet
    string      name;
    string      type; //HTML, VRML, Applet, FX
    string      url; //locator
25  long        size; //file size (for download monitoring)
    PrComponentVec & componentVec; //list of components (images, sound, video)
};

class PrComponent
30 {
    //image, video, sound, applet and other non-text components of a page
    string      name;
    string      mime_type; //image, sound, etc.
    long        size; //file size
35  long        playTime; //play (display) time; for timing
    string      url; //locator
};

class PrFx
40 {
    //special effect (filler to smooth out the presentation at changing conditions)
    string      name;
    string      type; //text msg, applet, animation clip
    string      selectionType; //random, by priority, specific
45  PrPageVec & pageVec; //page containing this FX
};

class PrEnvironment
50 {
    //class of browser environment
    //for which sequences can be created
    string      name;
55  };

#include "VauyWebPresSmooth.h"

60 class PresentationManager : public Applet
    {
        int ProbePage()
        {
65         //spawn a separate thread to gauge throughput rate
            spawnThread( "ThroughputTest");
        }
    }

```

```

5 //get maximum number of simultaneous connections
//the browser is allowed to open
maxConnectionCnt = Browser.getMaxConnectionCnt();

//get browser's buffer size
netBufferSize = Browser.getNetBufferSize();

10 //inquire what kinds of plugins the browser has
Browser.getPlugins(pluginVec);
for( plugin = pluginVec.firstElement;
    plugin != NULL;
    plugin = plugin.nextElement())
15 {
    switch plugin.type
    {
    case pluginStreamingVideo:
        streamVideoType = StreamVideoTypeVec(plugin.vendor);
        break;
20     case pluginVirtualReality:
        vrPlayer = Yes;
        break;
    } //switch
25 }

// wait for completion of the throughput test thread
waitFor( finish, "ThroughputTest");

30 //now we have all parameters needed to determine
//which of the presentation environment to choose
environ = determineEnvironment();

35 //preload the top-level menu, and display it
spawnThread( "PreloadPage", presView.rootMenuPage);
curPage = presView.rootMenuPage;
displayPage( curPage);

40 //determine which sequences will potentially be used
//in the presentation, as well as starting sequence
rootPresentation.startSequence = DetermineSequences( sequenceVec);

//preload first page of the chosen sequence
45 spawnThread( "PreloadPage", "Fx", presView.rootPresentation);

//spawn controlling thread
spawnThread( "rootSequenceManager");

50 //spawn a separate thread to monitor throughput rate
spawnThread( "PeriodicThroughputTest");

} //ProbePage()

55 int Presentation::SequenceManager()
{
    curSequence = startSequence;
    curEnv = startEnv;

60 //go through all items of the sequence
seqitem = mainSequence.pageVec.firstElement;
while( true)
65 {
    //reassess the state of the network (since prev. page)

```

```

curState = assessState();
switch curState
{
5   case stThroughputDidNotChangeDramatically:
      //can we preload one more component?
      if( preloadedCount >= MaxPreloadedCount)
      {
          break;
      }
10  nextPreloadSeqItem =
      curSequence.getNextPreloadSeqItem();
      nextPreloadComp =
      curSequence.getNextPreloadComponent();
15  //check if this comp is already cached
      nextPreloadComp.calcDownloadTime( curEnv);
      nextDownloadOK = true;

      //for each comp being downloaded,
      //see if starting new download will thwart timing
20  for(i = 0; i < MaxPreloadedCount; ++i)
      {
          comp = componentBeingDownloaded( i);
          if( comp.status != stInProgress)
          continue;
25  comp.calcTimeToStart();
          comp.calcRemainingTimeToDownload();

          if( comp.timeToStart <
30  comp.remainTimeDL +
          nextPreloadComp.totTimeDL)
          {
              //timing = Late; it is not OK to download
              nextDownloadOK = false;
              break;
          }
35  }
          if( !nextDownloadOK)
              break; //switch
          //OK to download this component; do it
          spawnThread( nextPreloadComp.preload);
          break; //switch

40  case stUrlNotFound: //file not found
      //can we skip this page?
      if( nextPreloadSeqItem.skipAllowed)
      //yes: skip it
      nextPreloadSeqItem.setSkip();
      else
50  //no: try to switch sequences
      nextPreloadSeqItem.setJumpSequence();

      case stThroughputIncreased:
          //keep count of how steady is the increase
          ThroughputIncreasedCount++;
          ThroughputDecreasedCount = 0;
          nextSequence.nextItem = curSequenceItem.upSeqItem();
          //can we afford (timing-wise) to go up the sequence
          nextSequence.nextItem.calcTiming();
          if(nextSequence.nextItem.timingOK)
60  //yes: switch
          setNextItem(nextSequence.nextItem);
          else
          //no: act as if no change
          setStatus(stNormal);
65

```

```

        break;

    case stThroughputDecreased:
5       //keep count of how steady is the decrease
        ThroughputDecreasedCount++;
        ThroughputIncreasedCount = 0;

        calcCurTiming();
10      if (curSequenceItem.timingOK)
            //we are still OK for this and next page
            break;

        //try different ways to handle oncoming delay
        //(returns success if we can handle it)
15      if ( (status = handleLateTiming()) == Success)
            break;

        //cannot handle delay with any other method
        //try to switch to lower sequence
20      curSequence = curItem.lowSequence;
        nextItem = curItem.getLowSeqItem();

        break;

    case stThroughputLow:
        //keep count of how steady is the decrease
        ThroughputDecreasedCount++;
        ThroughputIncreasedCount = 0;
30      calcCurTiming();
        if (curSequenceItem.timingOK)
            //we are still OK for this and next page
            break;
        else
        {
35          setStatus(stThroughputDecreased);
          continue; //next iteration picks it up and handles
it
        }
    } //switch
} //while
} //manager()

int handleLateTiming()
45 {
    //if we currently are displaying text,
    //increase viewing time by 20%
    //(whether it helps or not)
    if (curItem.page.mimeType = "text")
50 {
        increaseViewTime(curItem, percentIncreaseForText);
        calcCurTiming();
        if (curSequenceItem.timingOK)
        {
55          return Success;
        }
    }
}

60 //can we replay any components already loaded?
if ( curItem.replayCompVec != NULL)
{
    //rule: if comp i is replayed,
    //then i+1, i+2, ... must be replayed
65    comp = curItem.replayCompVec.lastElement();

```

```

5         while (comp != NULL)
        {
            virtualAddComp( comp);
            calcCurTiming();
            if(curSequenceItem.timingOK)
            {
                curItem.setReplay(comp);
                return Success;
            } else
            {
                //try one more replay-components
                comp = comp.PreviousElement();
            }
        }
        //are we preloading two pages ahead?
        //see if stopping that will help
        //if it does not help, do NOT stop preloading it yet!
        while(1)
        {
            seqItem = getLatestItemForWhichCompBeingPreloaded();
            if( seqItem != curItem
                && seqItem != curItem.nextElement())
            {
                virtualStopDownload(seqItem.downloadingComponents);
                calcCurTiming();
                if(curSequenceItem.timingOK)
                {
                    //we will be OK for this and next page :
                    stopDownload(seqItem.downloadingComponents);
                    return Success;
                }
            } else
            {
                break;
            } //while
        }
        //can we skip next page (or couple)?
        while( curItem.nextElement.skipAllowed)
        {
            curItem.nextElement.setSkip();
            calcCurTiming();
            if(curSequenceItem.timingOK)
            {
                //we will be OK for this and next page
                return Success;
            }
        }
        //try FXs available
        if( curItem.FxVec != NULL)
        {
            fx = curItem.FxVec.firstElement();
            while( fx != NULL)
            {
                virtualApplyFx( fx);
                calcCurTiming();
                if(curSequenceItem.timingOK)
                {
                    //we will be OK for this and next page
                    return Success;
                }
            }
        }
        return Fail;
    } //handleLateTiming()
} //class

```

CLAIMS

What is claimed is:

- 5 1. A computer based method for generating a web site having a plurality of web pages, the computer having a storage module and having a wide area network interface, the method comprising the steps of:
- designing a menu web page having one or more menu items;
- designing a sequence of web pages, including a first and second web page, for each of
- 10 said menu items, said first web page only capable of accessing one of said second web page and said menu web page, said second web page only capable of accessing the immediately subsequent web page in said sequence of web pages and said menu web page, said sequence of web pages:
- identifying each menu item with one of said sequence of web pages; and
- 15 storing said menu web page and web page sequence information in the storage module.
2. The method of claim 1, wherein said sequence of web pages is designed by a webmaster, said sequence of web pages enables the webmaster to control the content and sequence of information displayed to the client.
- 20 3. The method of claim 2, further comprising the step of:
- receiving a request signal from a destination computer via the wide area network interface, for requesting access to the web site;
- transmitting said menu web page and said web page sequence information to said
- 25 destination computer via the wide area network;
- displaying said menu web page to a client on a display device coupled to said destination computer;
- preloading additional web pages before said client selects a menu item;
- receiving a menu item selection from a client;
- 30 preloading said sequence of web pages associated with said selected menu item, using a dynamic preload algorithm; and
- displaying said sequence of web pages to the client without permitting the client to alter said sequence.

4. The method of claim 3. wherein said step of preloading said sequence of web pages includes the steps of:

determining a data receiving rate of the destination computer;

5 determining the size of two or more web pages in said sequence of web pages;

setting a priority value for each of said two or more web pages in said sequence of web pages based upon said data receiving rate, said size of said web page, and the position in said sequence of said web page;

10 preloading one of said two or more web pages into said destination computer storage module, based upon said priority value; and

dynamically modifying said priority value of one or more web pages to reduce any delay in displaying one of said web pages to the client.

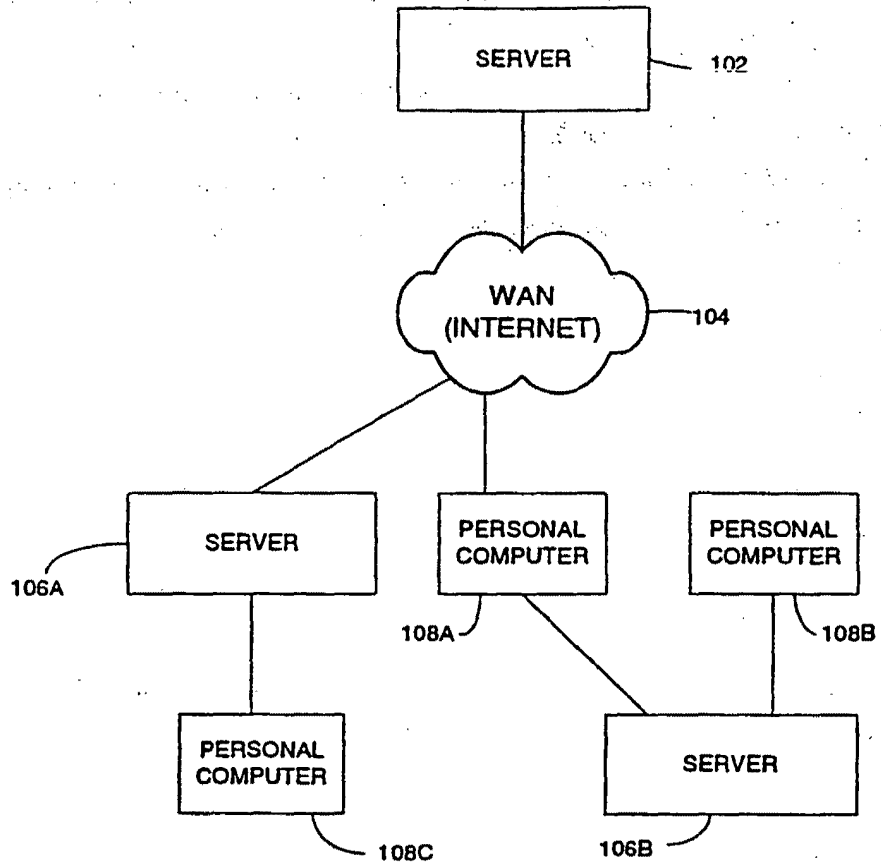


FIGURE 1

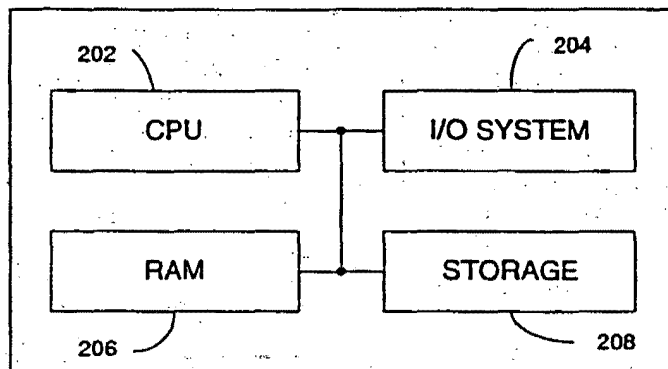


FIGURE 2

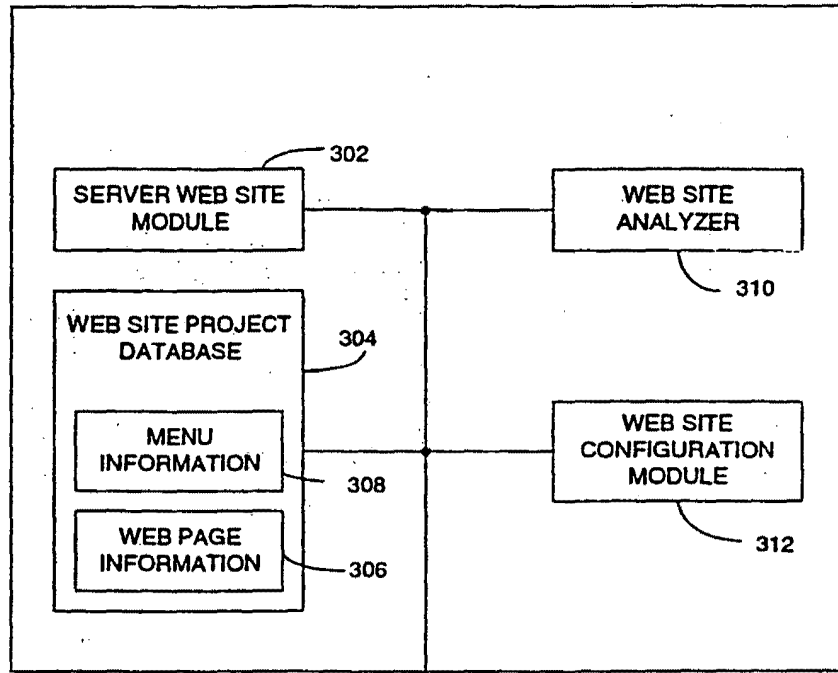


FIGURE 3

4/20

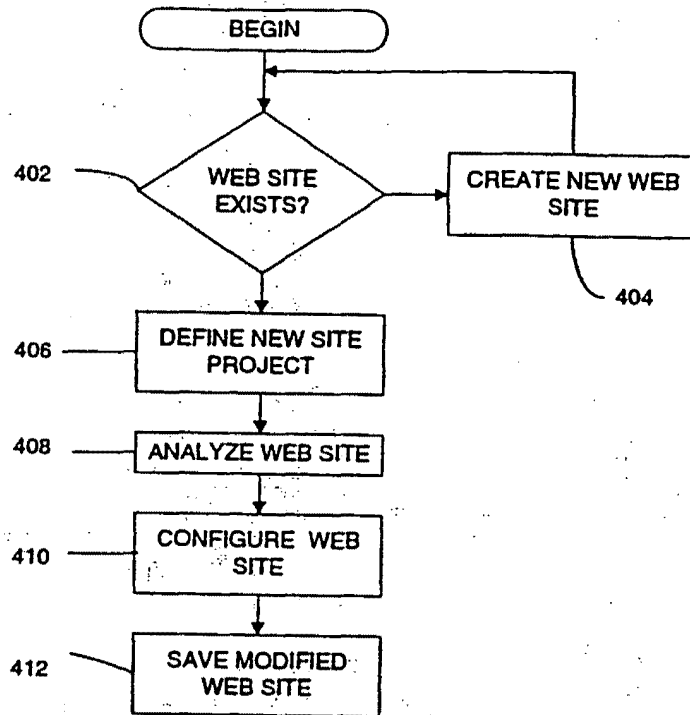


FIGURE 4

5/20

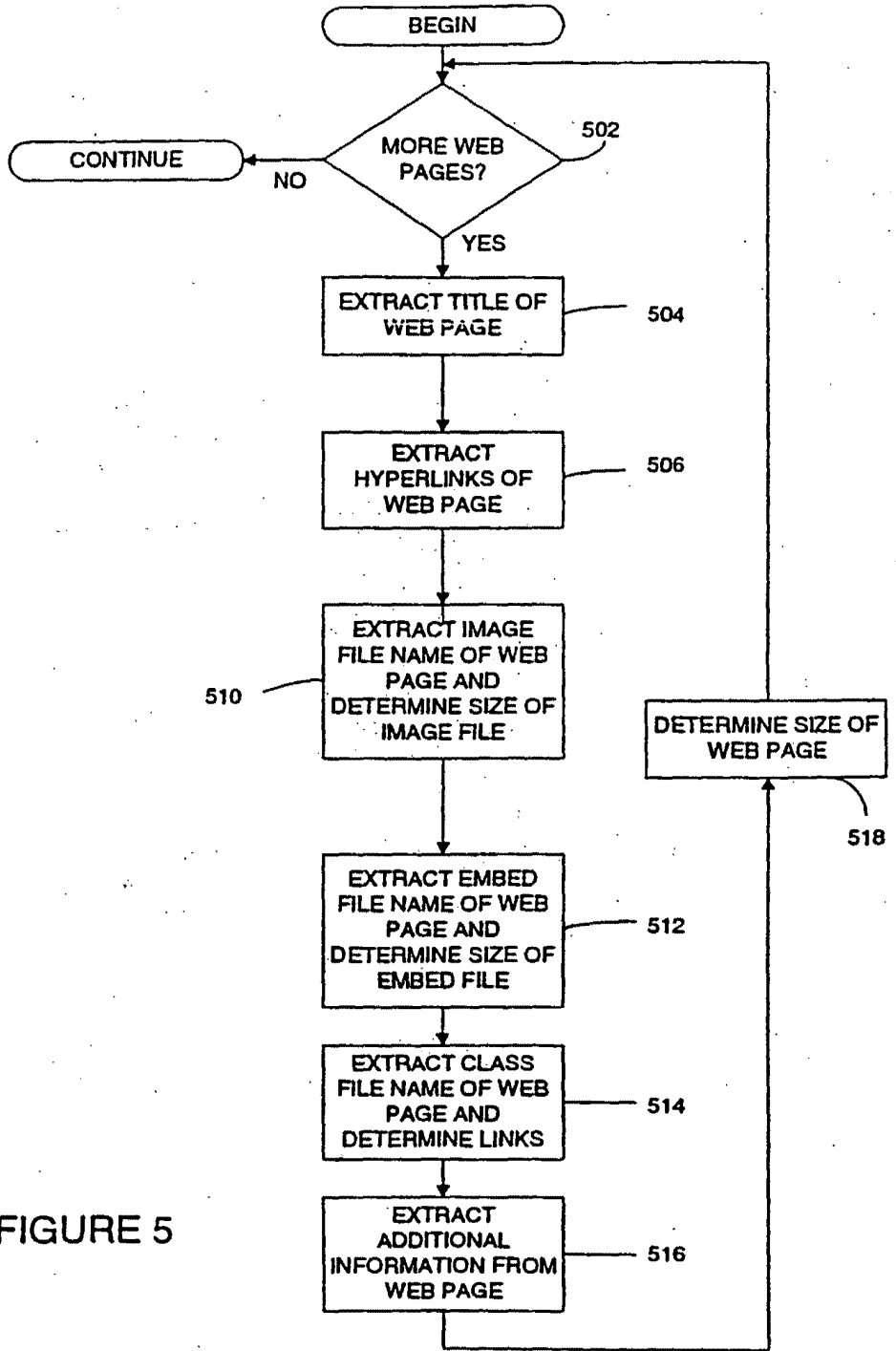


FIGURE 5

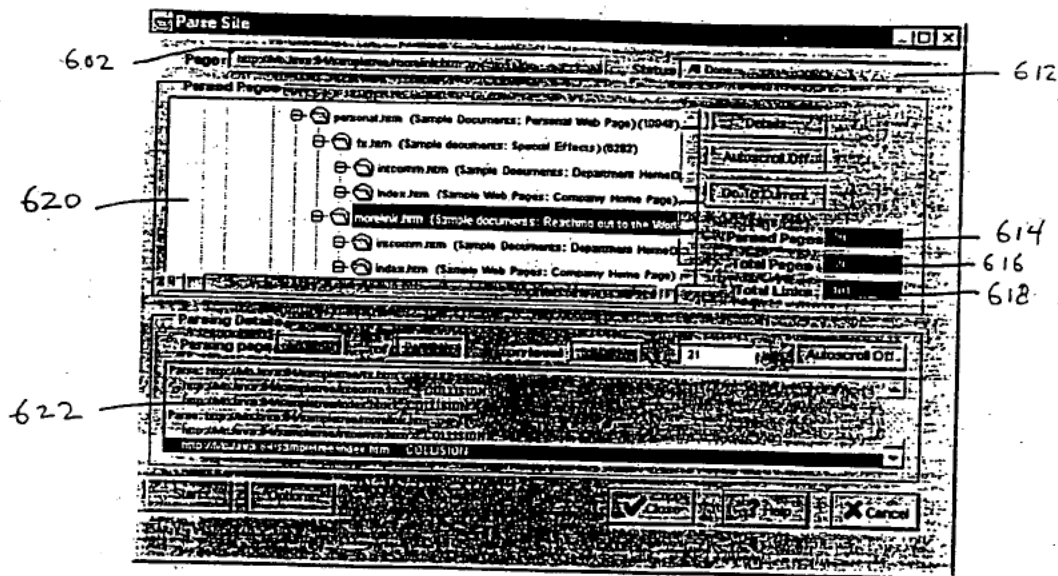
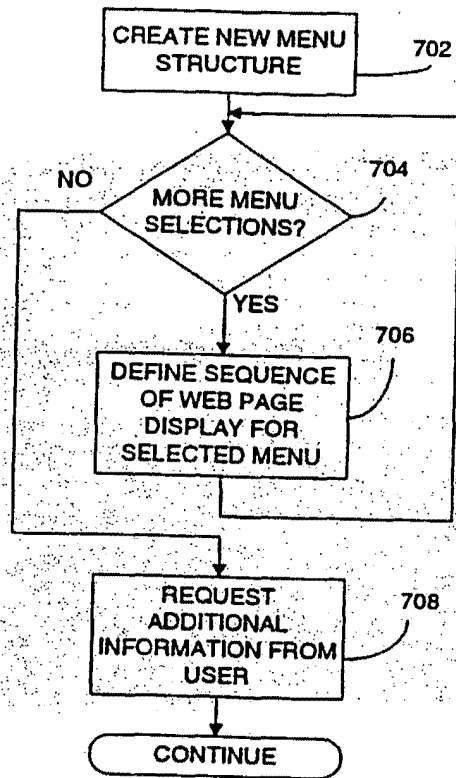


Figure 6

7/20



410

FIGURE 7

8/20

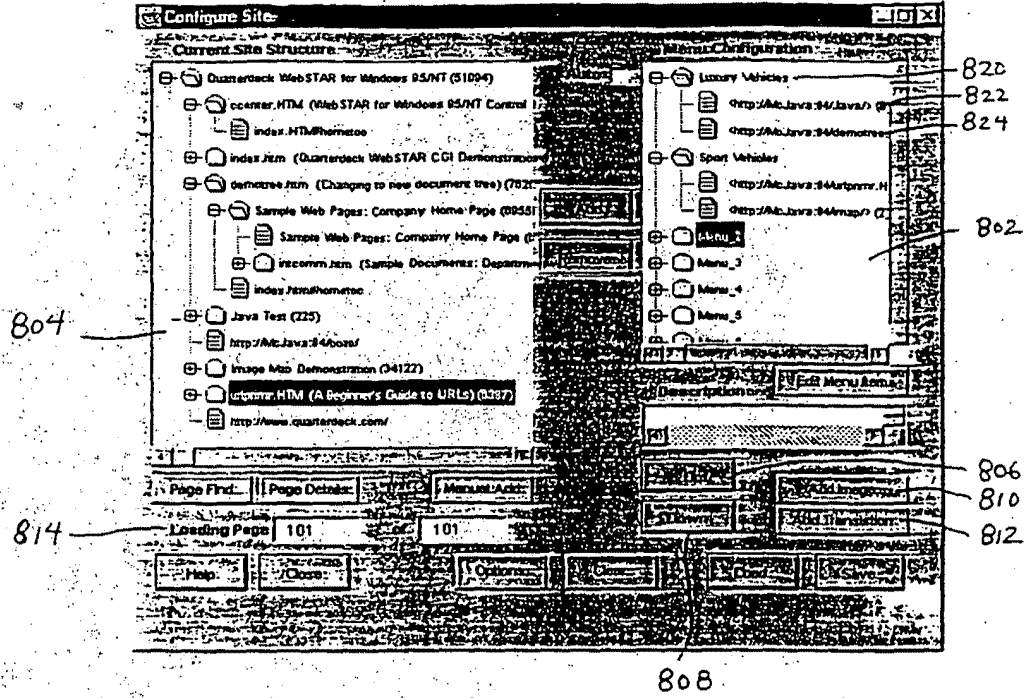


Figure 8

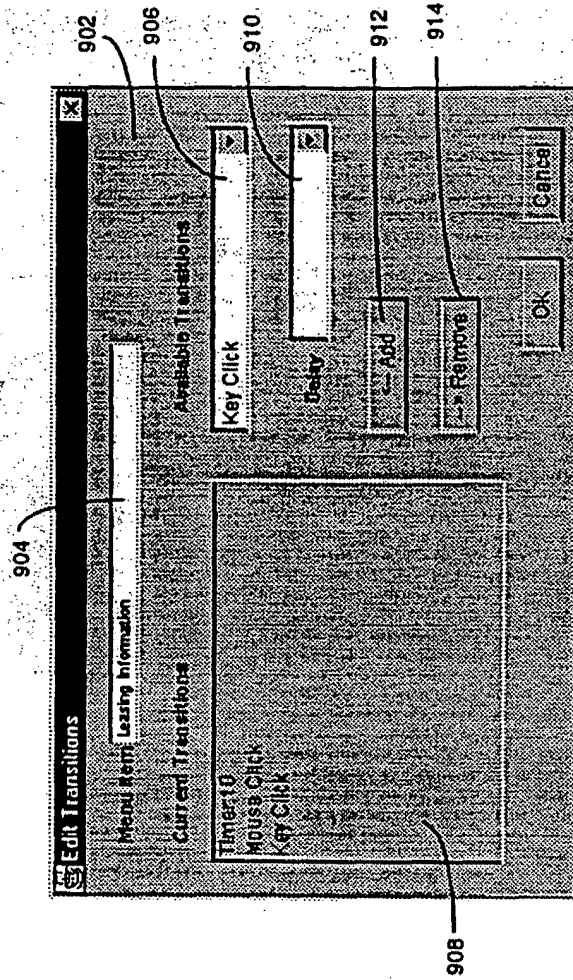


Figure 9

10/20

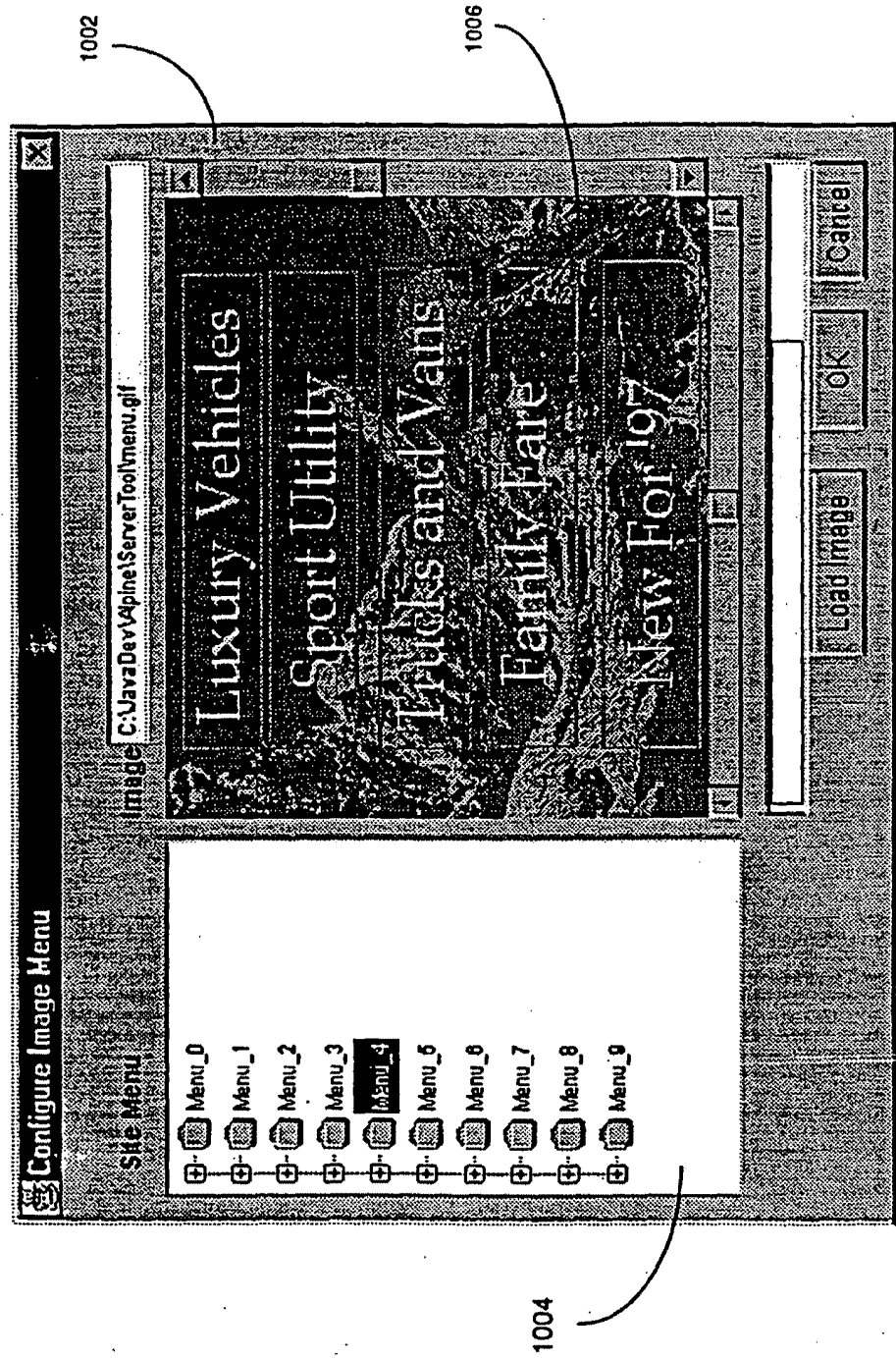


Figure 10

11/20

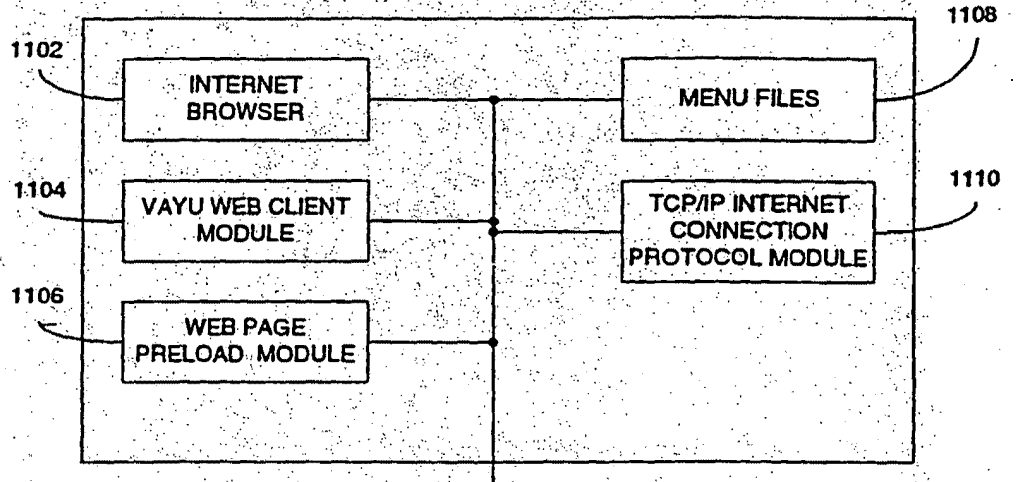


FIGURE 11

12/20

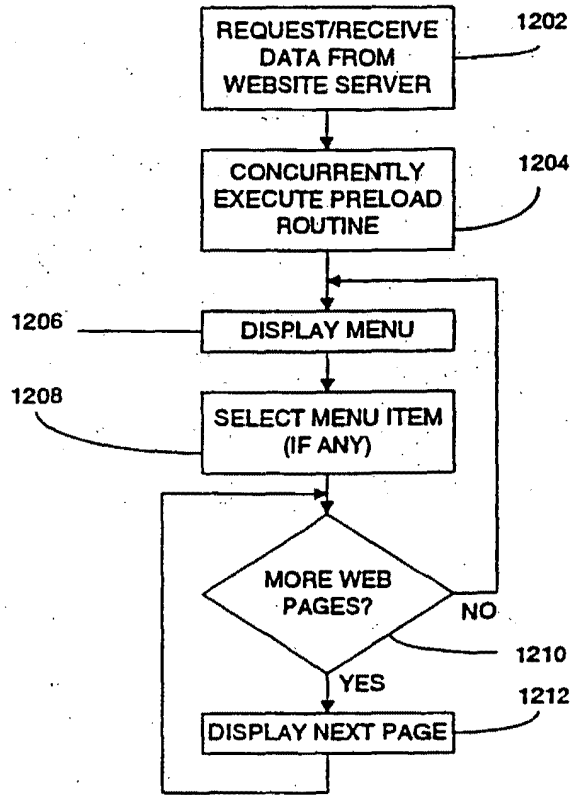


FIGURE 12

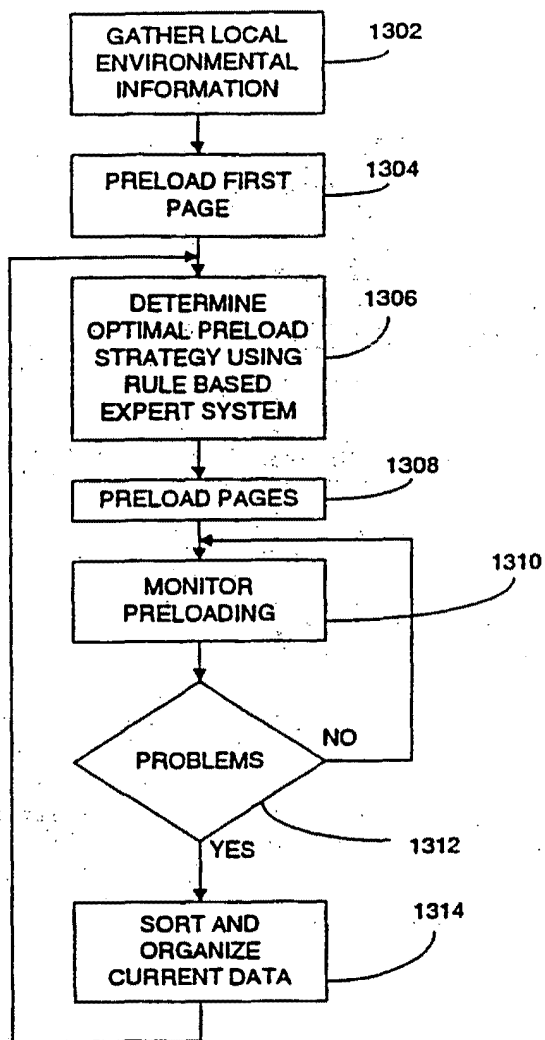


FIGURE 13

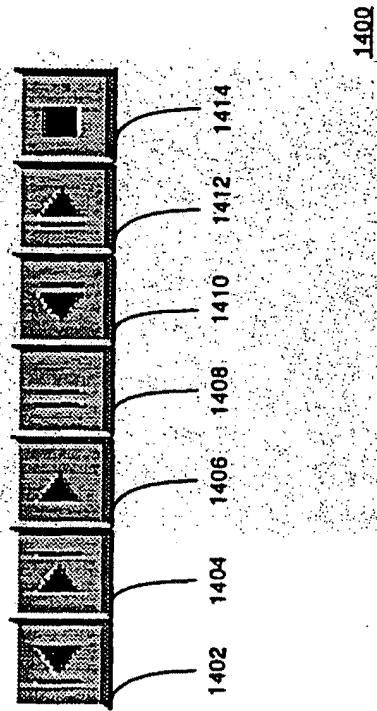


Figure 14

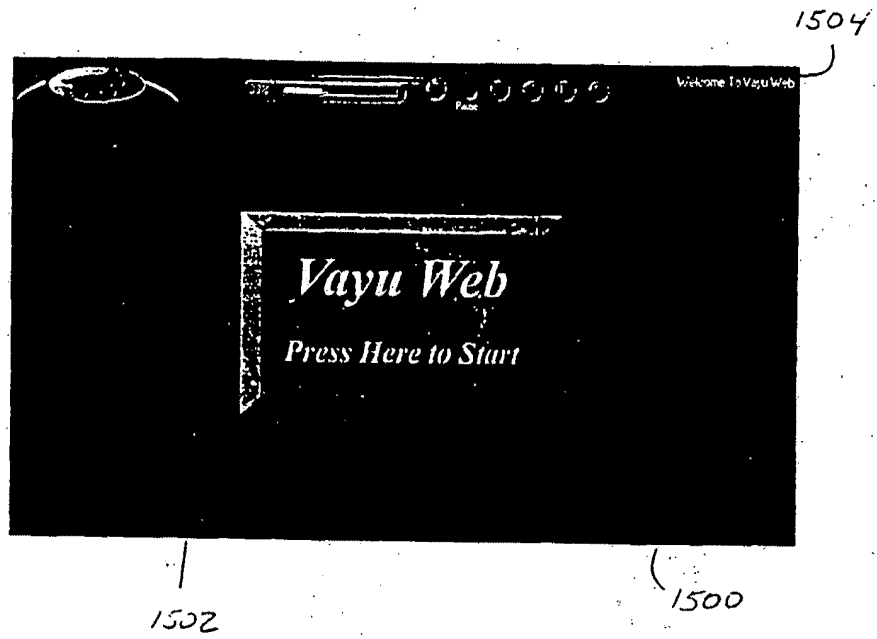


FIGURE 15

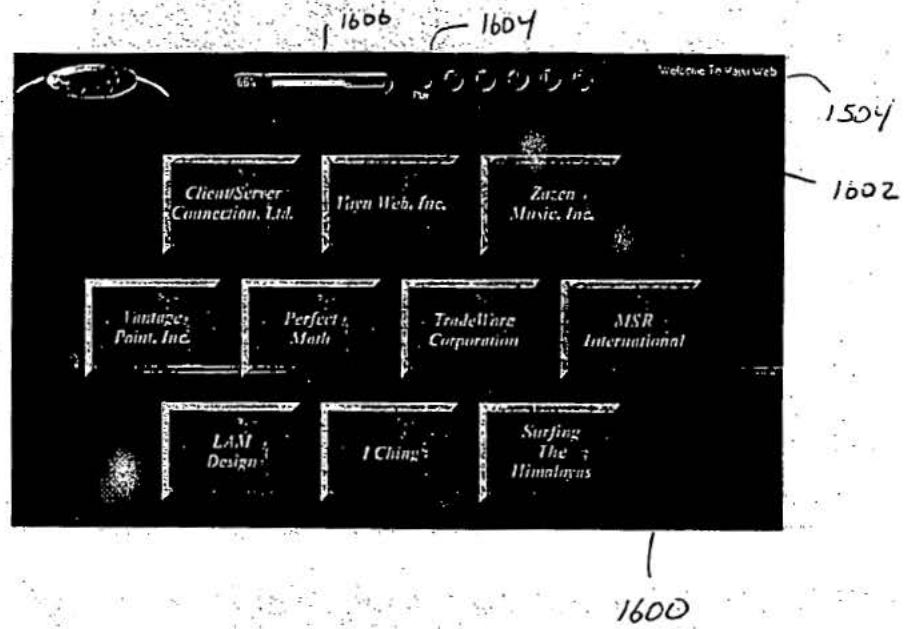


FIGURE 16

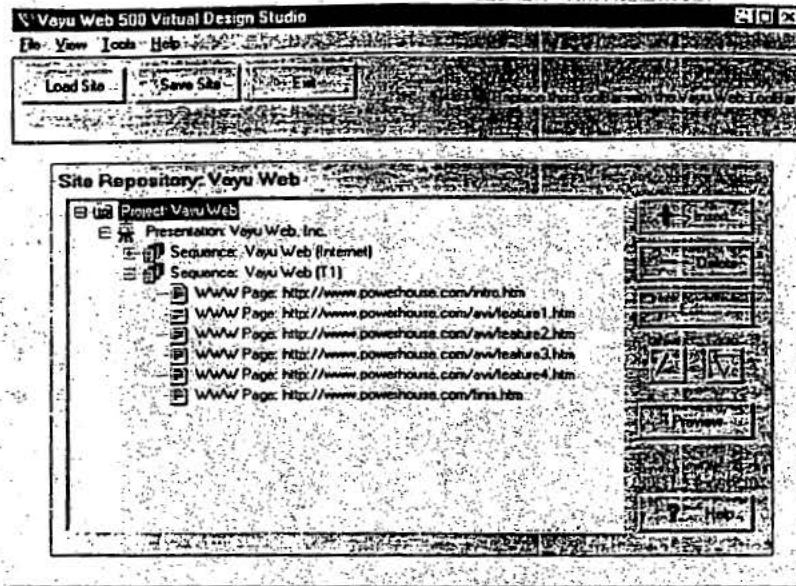


FIGURE 17

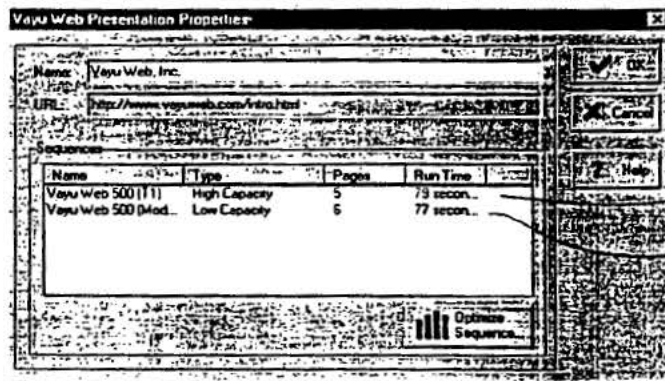


FIGURE 18

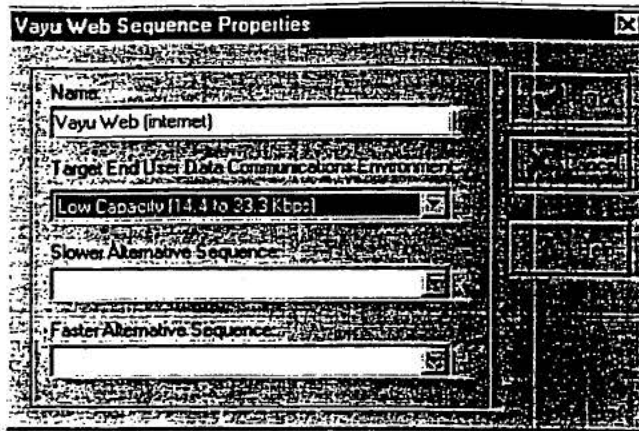
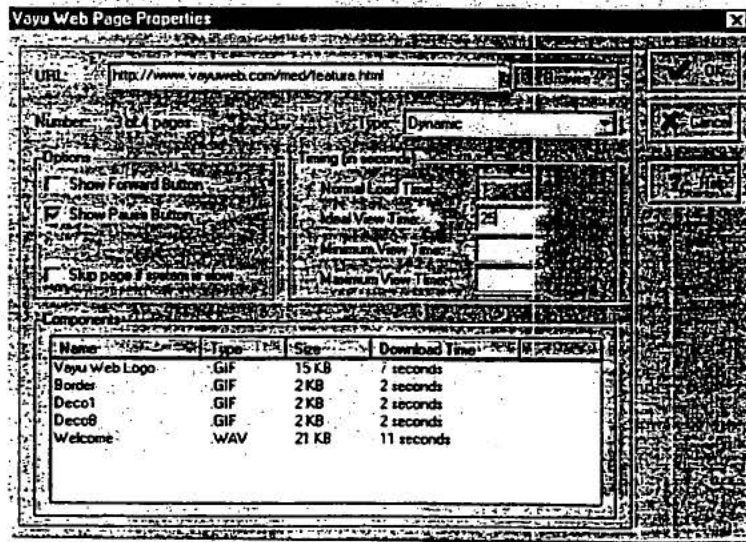


FIGURE 19

1900



2000

FIGURE 20

FIGURE 21

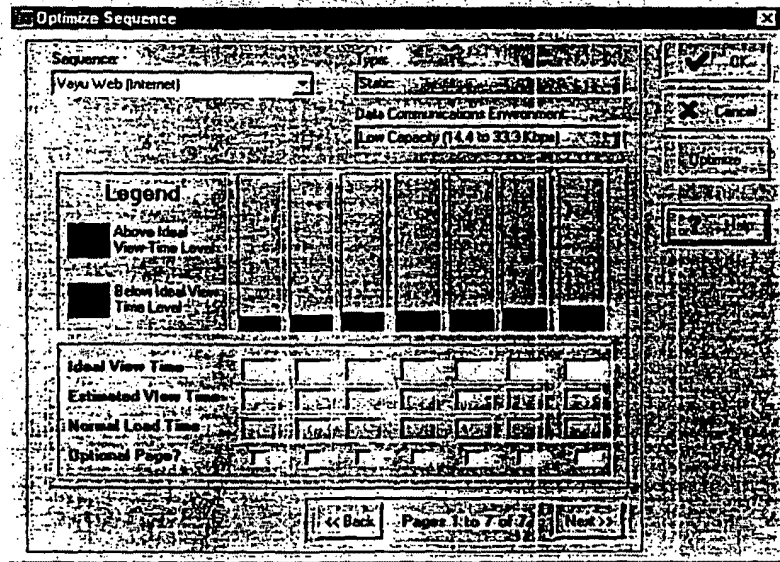
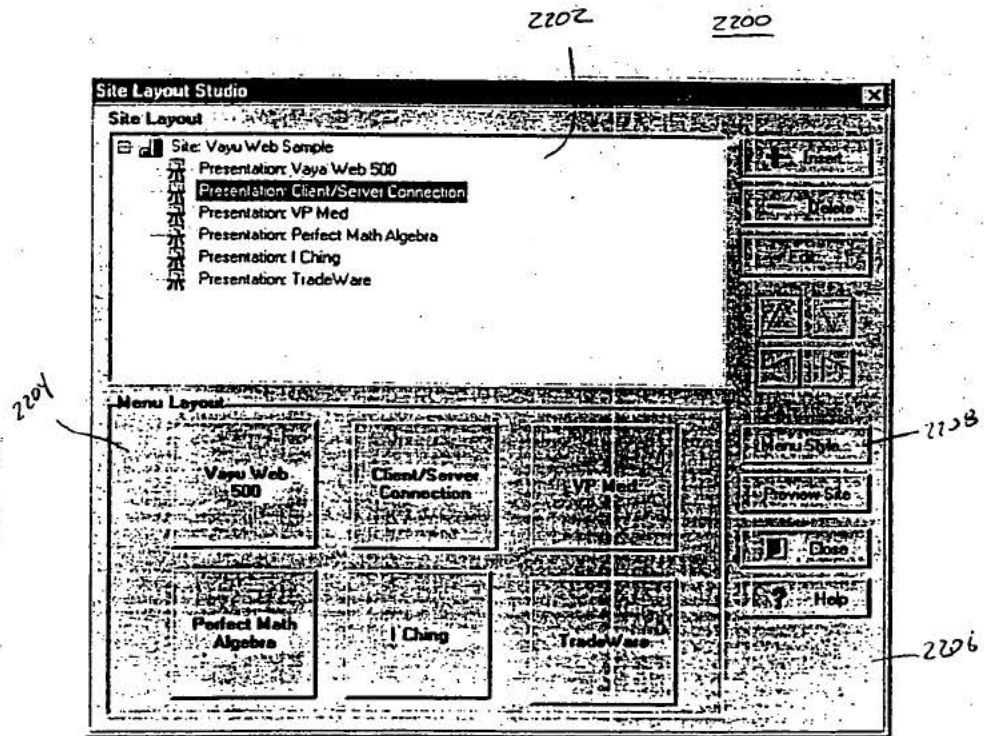
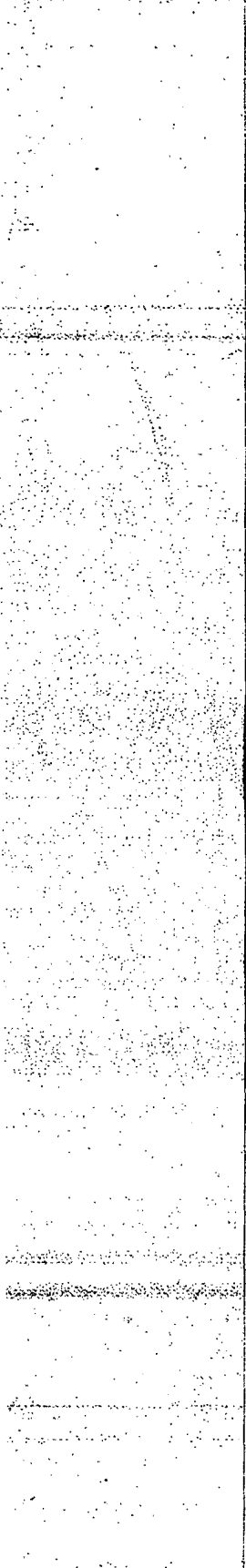


FIGURE 22





2025



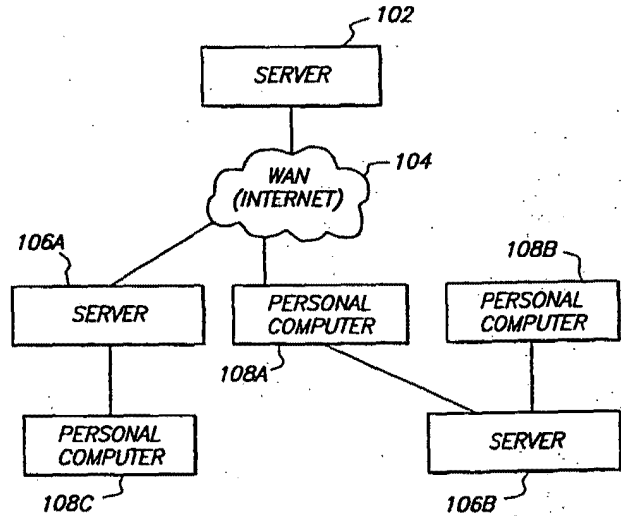
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification ⁶ : G06F 17/30</p>	<p>A3</p>	<p>(11) International Publication Number: WO 98/20434 (43) International Publication Date: 14 May 1998 (14.05.98)</p>
<p>(21) International Application Number: PCT/US97/19719 (22) International Filing Date: 30 October 1997 (30.10.97) (30) Priority Data: 08/745,899 7 November 1996 (07.11.96) US 08/762,289 9 December 1996 (09.12.96) US (71) Applicant: VAYU WEB, INC. [US/US]; Suite 206, 309 Mamaroneck Avenue, White Plains, NY 10605 (US). (72) Inventor: LENZ, Frederick, P.; 183 Old Field Road, Old Field, NY 11733 (US). (74) Agents: MCNELIS, John, T. et al.; Fenwick & West LLP, Suite 700, Two Palo Alto Square, Palo Alto, CA 94306 (US).</p>	<p>(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).</p> <p>Published With international search report.</p> <p>(88) Date of publication of the international search report: 20 August 1998 (20.08.98)</p>	

(54) Title: SYSTEM AND METHOD FOR DISPLAYING INFORMATION AND MONITORING COMMUNICATIONS OVER THE INTERNET

(57) Abstract

A system and method for creating, transmitting, and receiving web pages from a web server. The present invention enables a user to create a new web site by either creating new web pages or extracting information from previously created web pages. The invention enables a web site master to control the entire presentation of data to the user. In one embodiment of the present invention, a web site can include a first web page having a main menu. The present invention then permits the web site master to prepare a preset sequence of web pages that are displayed to the user for each menu item. The user of the web page selects one menu item from a menu display and then can passively receive information in a logical sequence. The control of the web page sequence can be used by advertisers, for example, to effectively present all information deemed helpful in attempting to convince the user to use the advertised product or service. In contrast to conventional web sites, a user accessing a web site developed according to the present invention will have a predefined sequence of web pages displayed in order to maximize the benefit to the web site master, e.g., to increase the behavioral modifications of the client due to controlled advertising. The present invention also increases the effectiveness of web sites having multiple web pages by using an artificial intelligence preloading methodology to reduce the delay in displaying a web page on the user's computer.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LJ	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

A. CLASSIFICATION OF SUBJECT MATTER IPC 6 G06F17/30		
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) IPC 6 G06F H04L		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practical, search terms used)		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	CRESPO A ET AL: "WebWriter: A browser-based editor for constructing Web applications" COMPUTER NETWORKS AND ISDN SYSTEMS, vol. 11, no. 28, May 1996, page 1291-1306 XP004018228 see page 1291, column 2, line 6 - page 1291, column 2, line 17 see page 1293, column 1, line 15 - page 1293, column 2, line 31 see figure 2	1,2
Y	---	3,4
	---	---
<input checked="" type="checkbox"/> Further documents are listed in the continuation of box C. <input checked="" type="checkbox"/> Patent family members are listed in 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 document 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 another 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. "Δ" document member of the same patent family
Date of the actual completion of the international search	Date of mailing of the international search report	
24 April 1998	04/05/1998	
Name and mailing address of the ISA European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Tx. 31.651 epo nl Fax: (+31-70) 340-3016	Authorized officer Abbing, R	

Form PCT/ISA/210 (second sheet) (July 1992)

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	PATENT ABSTRACTS OF JAPAN vol. 015, no. 476 (E-1140), 4 December 1991 & JP 03 204259 A (NIPPON TELEGR & TELEPH CORP), 5 September 1991, see abstract	3
Y	EP 0 669 587 A (AT & T CORP) 30 August 1995 see abstract see column 2, line 55 - column 3, line 18 see column 18, line 34 - column 21, line 13	4
X	LADD D A ET AL: "PROGRAMMING THE WEB: AN APPLICATION-ORIENTED LANGUAGE FOR HYPERMEDIA SERVICE PROGRAMMING" INTERNATIONAL WORLD WIDE WEB CONFERENCE, December 1995, pages 1-17, XP002049893 see page 6, line 24 - page 7, line 9	1,2

Form PCT/ISA/210 (continuation of second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No

PCT/US 97/19719

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 0669587 A	30-08-95	CA 2140850 A US 5715404 A	25-08-95 03-02-98

Form PCT/ISA210 (patent family annex) (July 1992)



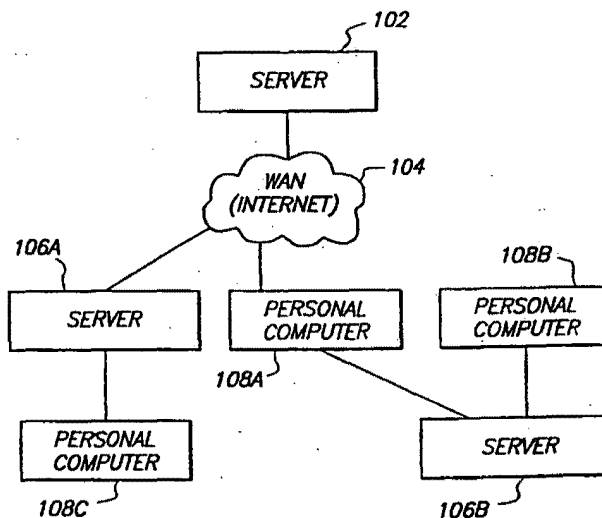
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification ⁶ : G06F 17/30</p>	<p>A3</p>	<p>(11) International Publication Number: WO 98/20434 (43) International Publication Date: 14 May 1998 (14.05.98)</p>
<p>(21) International Application Number: PCT/US97/19719 (22) International Filing Date: 30 October 1997 (30.10.97) (30) Priority Data: 08/745,899 7 November 1996 (07.11.96) US 08/762,289 9 December 1996 (09.12.96) US (71) Applicant: VAYU WEB, INC. [US/US]; Suite 206, 309 Mamaroneck Avenue, White Plains, NY 10605 (US). (72) Inventor: LENZ, Frederick, P.; 183 Old Field Road, Old Field, NY 11733 (US). (74) Agents: MCNELIS, John, T. et al.; Fenwick & West LLP, Suite 700, Two Palo Alto Square, Palo Alto, CA 94306 (US).</p>	<p>(81) Designated States: AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, GH, HU, ID, IL, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, UA, UG, UZ, VN, YU, ZW, ARIPO patent (GH, KE, LS, MW, SD, SZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG). Published With international search report. (88) Date of publication of the international search report: 20 August 1998 (20.08.98)</p>	

(54) Title: SYSTEM AND METHOD FOR DISPLAYING INFORMATION AND MONITORING COMMUNICATIONS OVER THE INTERNET

(57) Abstract

A system and method for creating, transmitting, and receiving web pages from a web server. The present invention enables a user to create a new web site by either creating new web pages or extracting information from previously created web pages. The invention enables a web site master to control the entire presentation of data to the user. In one embodiment of the present invention, a web site can include a first web page having a main menu. The present invention then permits the web site master to prepare a preset sequence of web pages that are displayed to the user for each menu item. The user of the web page selects one menu item from a menu display and then can passively receive information in a logical sequence. The control of the web page sequence can be used by advertisers, for example, to effectively present all information deemed helpful in attempting to convince the user to use the advertised product or service. In contrast to conventional web sites, a user accessing a web site developed according to the present invention will have a predefined sequence of web pages displayed in order to maximize the benefit to the web site master, e.g., to increase the behavioral modifications of the client due to controlled advertising. The present invention also increases the effectiveness of web sites having multiple web pages by using an artificial intelligence preloading methodology to reduce the delay in displaying a web page on the user's computer.



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MR	Mauritania	UG	Uganda
BY	Belarus	IS	Iceland	MW	Malawi	US	United States of America
CA	Canada	IT	Italy	MX	Mexico	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NE	Niger	VN	Viet Nam
CG	Congo	KE	Kenya	NL	Netherlands	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NO	Norway	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	NZ	New Zealand		
CM	Cameroon	KR	Republic of Korea	PL	Poland		
CN	China	KZ	Kazakhstan	PT	Portugal		
CU	Cuba	LC	Saint Lucia	RO	Romania		
CZ	Czech Republic	LI	Liechtenstein	RU	Russian Federation		
DE	Germany	LK	Sri Lanka	SD	Sudan		
DK	Denmark	LR	Liberia	SE	Sweden		
EE	Estonia			SG	Singapore		

18 minutes to download by the typical home computer, and only if the server that is sending the data is transmitting the data at the maximum permitted rate. While the data is being downloaded, conventional web sites display partial information on the screen of the users display monitor.

5 One growing use of the Internet and WWW is the creation of web sites for advertising. Advertisers are drawn by the low cost of the WWW. However, a significant drawback for advertisers is the slow rate of data transfer to typical home users. If an advertiser creates a web site where one page of the web site has a large size, e.g., two megabytes, then, as described above, a typical user having a 14.4 kbps modem will require over 18 minutes to download the
10 single web page. A significant number of users will be reluctant to download such large advertisement web pages due to the large time commitment.

 Another significant drawback for advertisers is that conventional web sites require the user to search hypertext and hypergraphics in each web page in order to proceed to what the user thinks is of interest. Therefore, an advertiser operating a web site having several pages
15 frequently has users browse the web pages in a haphazard manner. However, successful advertisers prefer more control of the presentation sequence of an advertisement.

 Accordingly, what is needed is a system and method for operating a web site that: (1) decreases the complexity of current web sites to reduce the intimidation of less sophisticated users of the web site; (2) decreases the waiting time between web page displays without
20 increasing the data transmission rate; and (3) enables a web site owner to have more control over the presentation of web pages to a user.

SUMMARY OF THE INVENTION

 The present invention is a new model for creating, transmitting, and receiving web pages
25 from a web server. The present invention enables a user to create a new web site by either creating new web pages or extracting information from previously created web pages. The invention enables a web site master to control the entire presentation of data to the user. In one embodiment of the present invention, a web site can include a first web page having a main menu. The present invention then permits the web site master to prepare a preset sequence of
30 web pages that are displayed to the user for each menu item. The user of the web page selects one menu item from a menu display and then can passively receive information in a logical sequence. The control of the web page sequence can be used by advertisers, for example, to effectively present all information deemed helpful in attempting to convince the user to use the

advertised product or service. In contrast to conventional web sites, a user accessing a web site developed according to the present invention will have a predefined sequence of web pages displayed in order to maximize the benefit to the web site master, e.g., to increase the behavioral modifications of the client due to controlled advertising.

- 5 The present invention also increases the effectiveness of web sites having multiple web pages by using an artificial intelligence preloading methodology to reduce the delay in displaying a web page on the user's computer.

BRIEF DESCRIPTION OF THE DRAWINGS

- 10 Figure 1 is an illustration of a communication system in which the preferred of the present invention resides.

 Figure 2 is a more detailed illustration of the server of the preferred embodiment of the present invention.

- 15 Figure 3 is a more detailed illustration of the server storage device of the preferred embodiment of the present invention.

 Figure 4 is a flow chart describing the method performed by the server module according to one embodiment of the present invention.

 Figure 5 is a flow chart describing the step of analyzing a web site according to one embodiment of the present invention.

- 20 Figure 6 is an illustration of a computer screen display showing information extracted from the web pages of the web site during the analyze web site 408 process according to one embodiment of the present invention.

 Figure 7 is flow chart describing the step of configuring a modified web site according to one embodiment of the present invention.

- 25 Figure 8 is an illustration of a computer screen display for use in the web site configuration process according to one embodiment of the present invention.

 Figure 9 is an illustration of an edit transition display according to one embodiment of the present invention.

- 30 Figure 10 is an illustration of a configuration image menu display according to one embodiment of the present invention.

 Figure 11 is an illustration of a client computer storage device according to one embodiment of the present invention.

Figure 12 is a flow chart describing the method performed by the Vayu Web client module according to one embodiment of the present invention.

Figure 13 is a flow chart describing the method performed by the web page preload module according to one embodiment of the present invention.

5 Figure 14 is an illustration of a web page control panel according to one embodiment of the present invention.

Figure 15 is an illustration of a first menu web page according to an alternate embodiment of the present invention.

10 Figure 16 is an illustration of a second web site menu page according to an alternate embodiment of the present invention.

Figure 17 is an illustration of a site repository information according to an alternate embodiment of the present invention.

15 Figure 18 is an illustration of a screen display of the web site presentation properties of various web site sequences developed by the web site designer according to an alternate embodiment of the present invention.

Figure 19 is an illustration of a web page sequence properties screen display according to an alternate embodiment of the present invention.

Figure 20 is an illustration of a web page property screen display according to an alternate embodiment of the present invention.

20 Figure 21 is an illustration of a sequence optimizer screen display according to an alternate embodiment of the present invention.

Figure 22 is an illustration of the site layout screen display that is displayed to the web site designer according to an alternate embodiment of the present invention.

25 DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

A preferred embodiment of the present invention is now described with reference to the figures where like reference numbers indicate identical or functionally similar elements. Also in the figures, the left most digit(s) of each reference number correspond(s) to the figure in which the reference number is first used.

30 Figure 1 is an illustration of a communication system in which the preferred embodiment of the present invention resides. The communication system includes a web server 102, a wide area network 104, destination servers 106A, 106B, and personal computers 108A-C. The personal computers 108A-C are conventional personal computers, e.g., an Apple computer

or a Compaq computer. The personal computers 108 can be independent or part of a network, for example. The personal computers 108 can be connected directly to the WAN, using a modem, for example, or can be connected to the WAN 102 via a server 106A that uses a leased line, for example. The server 106A can operate using a conventional operating system, for example, UNIX or Windows NT. In order for a user of a computer, e.g., destination computer 108A to access data representing world wide web pages (web pages) from a web server 102, the user of the personal computer 108A identifies a uniform resource locator (URL) address to a web page in the web server 102 and transmits a request signal to the WAN 104. The WAN 104 is conventional, e.g., the Internet, and can use routers to route the request signal to the web server 102. A conventional web server receives a request and transmits the requested web page to the destination computer 108A. The destination computer 108A receives the web page data and displays the web page. This process can repeat several times for each web page requested by the user. The multiple requests per page are to obtain each multimedia object embedded in the page, e.g., pictures, sounds, and videos.

The present invention is a new model for creating, transmitting, and receiving web pages from a web server 102. The invention includes a server module and a client module. The server module is stored and implemented by the web server 102. The client module is stored in the web server 102 and is transmitted from the web server 102 to the destination computer 108A after the web server 102 receives a request from the destination computer 108A. The client module is executed by the destination computer 108A when displaying the web pages of the web site. A more detailed discussion of the server module and the client module is set forth below.

Figure 2 is a more detailed illustration of the web server 102 of the preferred embodiment of the present invention. As stated above, the web server is conventional and includes a conventional central processing unit 202, e.g., a Pentium Pro processor that is commercially available from Intel Corporation, Santa Clara, California, a conventional input/output system 204, conventional random access memory 206, and a conventional storage device 208 having unconventional computer application programs stored therein. The computer application programs stored in the storage device 208 are described in detail below.

Figure 3 is a more detailed illustration of the web server storage device 208 of the preferred embodiment of the present invention. The storage device 208 includes a server web site module 302, a web site project database 304 having menu information 308 and web page information 306, a web site analyzer 310, and a web site configuration module 312. The server

web site module 302 includes web site data including MIME files. MIME files can include MIME types that provide a technique for enabling web browsers, e.g., Netscape Navigator, to automatically launch a viewing program associated with a received file, hypertext markup language (HTML) data, and common gateway interface (CGI) data. The web site project
5 databases 304 includes menu information 308 and data from web pages 306 extracted from the web page by the web site analyzer 310. A more detailed description of the web site project database is set forth below. The web site analyzer 310 analyzes each web page of the web site, extracts data from each web page, and stores the data in the web site project database 304. The web site configuration module 312 enables a user to create a menu and to create an ordered
10 sequence of web pages that can be presented to a user (client) when a menu item is selected.

Figure 4 is a flow chart describing the method performed by the server module 102 according to one embodiment of the present invention. If the desired web pages of a web site do not yet exist 402 then the web pages are created 404 using conventional web page creation techniques. A web site creation description is provided in Chandler, Running a Perfect Web
15 Site, Que Corporation (1995) that is incorporated by reference herein in its entirety. If the web pages that will be part of the web site exist 402 then the server 102 defines 406 a new site project, analyzes 408 the web pages of the web site, configures 410 the new web site, and saves 412 the web site. A more detailed description of steps 406, 408, 410, and 412 is set forth below. Alternatively, web pages for the web site can be developed concurrently using the present
20 invention.

As stated above, the web server 102 defines 406 a new site project. In one embodiment a new site project is a data structure that is stored in the server storage module 208. When fully defined, the new site project includes or references all of the information about a web site. The information stored about each web site includes general information about the web site and a list
25 of the web pages in the web site. The general information includes information such as: (a) the company's name where the company can be the party to which the web site information is directed, e.g., the company who is advertising on the web site; (b) the webmaster name, e.g., the creator of the web site; (c) the webmaster electronic-mail (e-mail) address. The list of web pages includes information about each web page, such as: (1) the URL of the web page; (2) a list
30 of links out of the page; (3) a list of links into the page from other web pages in the web site; (4) the title of the web page; (5) the content type of the page, e.g., text, html, and image; (6) a list of pictures/videos/sounds in the page; and (7) the size of the web page. The size (in bytes) of the web page can be determined using any of a variety of techniques. In one embodiment, the size

of the web page is the sum of the number of bytes in the page's HTML, the size of all image , sound <BGSOUND>, object <OBJECT>, applet <APPLET>, and script <SCRIPT> tag files. The information stored about each web page is determined during the steps of analyzing 408 the web site and configuring 410 the web site as described below.

5 Figure 5 is a flow chart describing the step of analyzing 408 a web site by the web site analyzer 310 according to one embodiment of the present invention. The web site analyzer determines the size of the web page and determines the links into and out of the page as described below. The web site analyzer 310 determines 502 if more web pages need to be analyzed. The web site analyzer 310 selects a web page by selecting one of the hyperlinks of a
10 previously analyzed web page. The web page analyzer 310 analyzes the web page and sets a flag indicating that the web page has been analyzed. If the web site analyzer 310 determines that the web page has already been analyzed, then the web site analyzer sets a pointer to the memory location in the server storage 208 containing the information previously extracted during the web page analysis.

15 The web page analyzer 310 extracts 504 the title of the web page. In one embodiment, the title of the web page is extracted by identifying the text associated with the title tag (<TITLE>). The title can be displayed in order to identify the web page to the user. The web page analyzer 310 then extracts 506 the hyperlinks of the web page. In one embodiment, the hypertext links are extracted by identifying the anchor tags (<A>), the frame tags (<FRAME>),
20 and map tags (<MAP>) in the web page. The web page analyzer 310 extracts 510 image file names by, for example, identifying the file associated with the image tag (), and the web page analyzer 310 determines the size of the image file by (a) reading the HTTP size header or (b) downloading the file. In one embodiment, the HTTP size header is read since it is typically faster than downloading the file. The web page analyzer 310 also extracts 512 embedded file
25 names in the web page by, for example, identifying the file associated with the embedded tag (<EMBED>). Then the web page analyzer 310 determines the size of each embedded file by (a) reading the HTTP size header or (b) downloading the file. The web page analyzer 310 also extracts 512 object files in the web page by, for example, identifying the file associated with the object tag (<OBJECT>). Then the web page analyzer 310 determines the size of each object file
30 by (a) reading the HTTP size header or (b) downloading the file. The web page analyzer 310 also extracts 512 sound files in the web page by, for example, identifying the file associated with the bgsound tag (<BGSOUND>). Then the web page analyzer 310 determines the size of each sound file by (a) reading the HTTP size header; or (b) downloading the file. The web page

analyzer 310 extracts 514 class file names by, for example, identifying the file names associated with a class file name tag, e.g., <APPLET> in the Java programming language commercially available from Sun Microsystems, Inc., Santa Clara, California. The web page analyzer then determines a list of links out of the web page by downloading and scanning the contents of each class file. The web page analyzer 310 also extracts 516 additional information from the web page. For example, the web page analyzer 310 identifies scripts, e.g., Javascripts and VBScripts, associated with a script tag (<SCRIPT>), downloads the scripts and scans the scripts for links out of the web page. After extracting the above identified information, the web page analyzer 310 determines 518 the size of the web page by summing the number of bytes in (1) the HTML of the web page, and (2) all of the image/embed/class/bgsound/object/script tag files. Steps 504-518 are repeated for each web page.

Figure 6 is an illustration of a computer screen display showing information extracted from the web pages of the web site during the analyze web site process 408, described above, according to one embodiment of the present invention. The computer display illustrated in Figure 6 displays web page information including the web site name 602 "http://McJava/sampletree/morelink.htm", and the status of the analysis 612, a current page identifier 614, the total number of pages 616, the total number of links found in all of the analyzed pages 618. In addition, the display includes information concerning the parsed pages 620 and parsing details 622. The parsed pages 620 shows the list of web pages as they are being parsed. The pages are displayed in a tree structure format that shows the hierarchical nature of the web site. The parsing details 622 shows a detailed list of the links and files found in the page being currently parsed. The site name 602 is the URL of the web site being analyzed. Host information (not shown) can also be displayed. The host is the name of the host machine on which the web site resides. Similarly, a tree level indicator (not shown) can also be displayed. The tree level represents the hierarchical level of the web page in the web site. Any web page may exist at a number of levels (e.g., it may appear as a hyperlink in many different pages in the web site). The level represented here is the level closest to the top of the tree that the page appears. It will be apparent to persons skilled in the art that more or less information can be available to the webmaster when configuring the web site.

After the web site analyzer 310 analyzes 408 the web site, the web site configuration module 312 configures the web site. Although the web site analyzer 310 has searched and analyzed each web page in the web site, the web site analyzer 310 does not modify the original web site. The web site configuration module 312 enables the web site developer, e.g., a

webmaster. to construct a menu. In one embodiment of the present invention this menu is typically approximately 8-10 items, although it is envisioned that any number of items can be part of the menu. Each item will consist of a sequence of web pages that are displayed in a fixed order. In one embodiment of the present invention the web pages themselves are not
5 modified except for the web page having the menu. In other embodiments the web pages are modified to, for example, ensure that all links out of the web page are deactivated, except for the links defined by the user. Figure 7 is a flow chart describing the step of configuring a modified web site by the web site configuration module 312 according to one embodiment of the present invention. The web site configuration module 312 enables the webmaster to create a menu
10 structure that can be the first web page displayed in the web site. For example, Figure 8 is an illustration of a computer screen display for use in the web site configuration process according to one embodiment of the present invention. The display includes a menu configuration portion 802, a current web site structure portion 804, a status portion 814, and various control buttons that can be selected by the webmaster using a conventional mouse or other pointing device. For
15 example. The control buttons in one embodiment include a menu shuffle up button 806, a menu shuffle down button 808, an add image button 810, an add transition button 812, and a status display 814.

The webmaster determines 704 if any more menu selections need to be modified. If the webmaster decides to modify or define a menu selection, the webmaster selects a menu, and
20 defines 706 a sequence of web pages that will be displayed when the menu is selected. One process for assigning a web page to a menu is: (a) a user highlights the menu in the menu configuration portion 802 using the mouse or keyboard; (b) the user then highlights the web page in current web site structure portion 804; and (c) the user selects the Add button 816. The chosen web page will be added to the menu and will be displayed in the menu configuration
25 portion 802 of the display. For example, a first menu is identified as "Luxury Vehicles" 820 in the menu configuration portion 802 of the display. The menu configuration portion 802 also includes information about the web pages that the webmaster has associated with the menu. For example, the top menu item is "Luxury Vehicles" 820. There are two web pages 822/824 associated with this menu item. These two web pages will be displayed in this order, i.e., web
30 page 822 and then web page 824, when the user ultimately views this web site with their client PC and browser.

The shuffle up button 806 and the shuffle down button 808 enable a user to modify the web site structure by moving a web page up or down in the displayed menu configuration

portion 802 of the display. These buttons simplify the web site editing procedure for the web master. For example, a web page can be selected by the webmaster and associated with one of the menu items. The webmaster can then move, add, delete, or copy web pages in order to develop a suitable web site presentation sequence. The webmaster can also introduce transitions that can control the transition between web pages by selecting the "Add Transition" button 812. For example, a transition may be a timer of 10 seconds, or a keyclick from the user. In one embodiment the webmaster ensures that the user will view the web pages associated with each menu (if any) only in the predefined sequence by automatically showing the next page in response the transition event. For example, if a timer of 10 seconds has been selected by the web master then after 10 seconds has elapsed the next page will automatically be displayed. The user does not need to perform any action. In alternate embodiments, a limited number of page links are included that enable the user to advance to the next page, return to the previous page, or return to the main menu, for example.

As noted above, conventional web sites are created such that a client who accesses the web site can view the various pages in the web site in almost any sequence. One potentially significant use of the Internet and the WWW is in advertising. In one embodiment, the present invention can be utilized to create a web site having advertisement as a significant purpose of the web site. For example, an automobile company, e.g., General Motors, can develop a web site that advertises its currently available cars and trucks. One important aspect of advertising is the process of creating a need that can be satisfied by the advertiser. In order to efficiently and successfully accomplish this, it is helpful if the advertiser is able to control the sequence of information presented to the user. For example, if a car manufacturer developed a web site, they may not want the client to immediately jump to the web page having the cost of the vehicles. Instead, the advertiser could be more successful and effective if a description of the features and benefits of one or more car models were first shown to the user in order to familiarize the client with the cars and to therefore help persuade the client that the cars offered by the manufacturer will satisfy the needs of the client. After providing the information to the user, the web page having pricing information can be presented to the user. It will be apparent to persons skilled in the art that the above example is merely one example of how the current invention could be used and that many alternatives exist for developing and strategically planning the web page presentation sequence in a web site. The present invention provides the necessary tools to enable a webmaster to quickly, efficiently, and effectively generate a web site that gives control of the web site presentation to the web site owner instead of the client.

The page link can be displayed as a "Next Page" button or the webmaster can select to have one or more of the web pages transition automatically. Figure 9 is an illustration of an edit transition display 902 according to one embodiment of the present invention. In the example illustrated in Figure 9 the user has the ability to define the event that will trigger a web page transition for any particular web page, e.g., the "Leasing Information" web page as displayed in the menu item display area 904. The available types of transitions are displayed in the available transition portion 906 of the display. The webmaster selects one of the available transition types and activates the Add button 912, and the selection is displayed in the current transition portion 908 of the display. If the webmaster selects the timer option, the webmaster can set the delay between pages by entering the delay in the delay definition portion 910 of the display. To remove a selection from the Current Transition 908 display the web master uses the Remove button 914.

The webmaster also has the opportunity to provide 708 additional information concerning the web site and the individual web pages. For example, some of the information stored in the new site project data structure is defined by the webmaster, e.g., information concerning the company name, webmaster name, webmaster e-mail, and the URL of the main menu page or one or more web pages in the web site, for example. The webmaster has the opportunity to provide this information before or after the defining 706 the web page display sequence for the web site.

Figure 10 is an illustration of a configuration image menu dialog box display 1002 according to one embodiment of the present invention. The dialog box 1002 enables the user to generate a menu display by selecting a menu, e.g., Menu_4, and defining the active regions in the display portion 1006 for the selected menu. In Figure 10, the active region for Menu_4 ("New for '97") is defined in the active region definition dialog box 1008. In some embodiments of the present invention the webmaster does not create an initial menu page, instead only one web page sequence is defined and when the client selects the first page the sequence of the remaining web pages of the web site are displayed to the client in the predefined sequence.

After the webmaster defines the menus and defines the web page sequence associated with each menu, the modified web site is saved 412. In one embodiment of the present invention, the web site is saved as a series of files that are used when web site information is transferred to a client accessing the web site. In one embodiment these files are used to transfer the information gathered by the server tool to the client. The files include a stub Java Applet that is inserted into the home page of the site. One example of such a file is set forth in Table 1.

```
<APPLET CODE="browser.class" WIDTH="10" HEIGHT="10">
5 <PARAM NAME="menu_1" VALUE="{MENU_ITEM NAME='Luxury
  Vehicles' AUTO=TRUE RECT=199,92,230,34 IMG=menu.gif}">
  <PARAM NAME="menu_1_1" VALUE="{LINK
    URL=http://www.xyzmotors.com/L2_N2.HTM SIZE=3514
    TRANSIT=TIMER DELAY=10}">
10     <PARAM NAME="menu_1_1_1" VALUE="{LINK OBJECT
      URL=http:// www.xyzmotors.com /Images\I_1_9.GIF
      SIZE=1059 CONTENT=image/gif}">
15     <PARAM NAME="menu_1_2" VALUE="{LINK_FILE
      NAME=xyzmotors_menu_1.brw}">
  <PARAM NAME="menu_2" VALUE="{MENU_ITEM NAME='Sport Utility'
    RECT=199,129,229,39 IMG=menu.gif}">
20     <PARAM NAME="menu_2_1" VALUE="{LINK URL=http://
      www.xyzmotors.com /L1_N1.HTM SIZE=12117
      TRANSIT=MENU}">
25     <PARAM NAME="menu_2_1_1" VALUE="{LINK OBJECT
      URL=http:// www.xyzmotors.com /Images\I_1_1.GIF
      SIZE=1257 CONTENT=image/gif}">
  <PARAM NAME="menu_2_2" VALUE="{LINK_FILE NAME=
30 xyzmotors_menu_2.brw}">
  <PARAM NAME="menu_3" VALUE="{MENU_ITEM NAME='Trucks and
    Vans' RECT=198,173,230,47 IMG=menu.gif}">
35     <PARAM NAME="menu_3_1" VALUE="{LINK URL=http://
      www.xyzmotors.com /L1_N1.HTM SIZE=12117 TRANSIT=TIMER
      DELAY=10}">
40     <PARAM NAME="menu_3_1_1" VALUE="{LINK OBJECT
      URL=http:// www.xyzmotors.com /Images\I_1_1.GIF
      SIZE=1257 CONTENT=image/gif}">
  <PARAM NAME="menu_3_2" VALUE="{LINK_FILE NAME=
45 xyzmotors_menu_3.brw}">
  <PARAM NAME="menu_4" VALUE="{MENU_ITEM NAME='Family Fare'
    RECT=195,224,232,40 IMG=menu.gif}">
50     <PARAM NAME="menu_4_1" VALUE="{LINK URL=http://
      www.xyzmotors.com /L1_N1.HTM SIZE=12117 TRANSIT=TIMER
      DELAY=10}">
55     <PARAM NAME="menu_4_1_1" VALUE="{LINK OBJECT
      URL=http:// www.xyzmotors.com /Images\I_1_1.GIF
      SIZE=1257 CONTENT=image/gif}">
  <PARAM NAME="menu_4_2" VALUE="{LINK_FILE NAME=
    xyzmotors_menu_4.brw}">
```



```

5      <PARAM NAME="menu_5" VALUE=" [MENU_ITEM NAME='New For
      '97'] ">
      <PARAM NAME="menu_5_1" VALUE=" [LINK URL=http://
      www.xyzmotors.com /L1_N1.HTM SIZE=12117 TRANSIT=TIMER
      DELAY=10] ">
      <PARAM NAME="menu_5_1_1" VALUE=" [LINK_OBJECT
10     URL=http:// www.xyzmotors.com /Images\I_1_1.GIF
      SIZE=1257 CONTENT=image/gif] ">
      <PARAM NAME="menu_5_2" VALUE=" [LINK_FILE NAME=
15     xyzmotors _menu_5.brw] ">
      </APPLET>

```

Table 1

20 These parameters in Table 1 contain specific information about each link on the menu illustrated in the display portion 1006 of Figure 10, and the contents of the pages linked thereto. The menu items are passed as parameters. In the example set forth in Table 1, a naming convention is used to determine the order and structure of the menu. The data is structured with a format that is similar to HTML, but uses square brackets ([and]) instead of angle brackets (< and >). The first parameter (PARAM) is for the first item in the menu, named *menu_1*. Its type is *MENU_ITEM*, and the item's name is *Sport Utility*. The next *PARAM* is a page assigned to the *menu_1*, and its name is *menu_1_1*. The type is *LINK*, its *URL* is <http://McJava/dealers/dealers.html>, the *TRANSIT* is *TIMER*, and the *DELAY* is 10 seconds. The third *PARAM* is also assigned to the *menu_1* item, but it points to a file that contains the rest of the links for *menu_1*. The file name is *McJava_menu_2.brw*. There are one of these files for each menu item that has more than one link assigned to it. These files are used to limit the amount of information that is transmitted from the server to the client in the initial transmission. Since the number of web pages that could be assigned to all of the menu items could range into the hundreds and thousands, it is generally more efficient to send down a portion of menu information, and then retrieve more menu information as needed. The file structure design is extensible such that as new information is needed new name/value pairs are added. Older version of the software will ignore name/value pairs they do not understand. Newer version of the software will have reasonable defaults if they encounter an older file with missing name/value pairs.

40 As described above, a client utilizes a computer, e.g., a destination computer 108A, to access the defined web site stored on the server 102. The personal computer used by the client

can be a conventional personal computer, e.g., an IBM personal computer that is commercially available from IBM Corporation, Armonk, NY. The destination computer 108A includes a conventional CPU, conventional RAM, and a conventional storage device, that can be similar to the server 102 described above. Figure 11 is an illustration of a client computer storage device 5 1100 according to one embodiment of the present invention. The storage device 1100 includes a conventional Internet browser 1102, a Vayu Web client module 1104, a web page preload module 1106, menu files 1108, and a TCP/IP Internet connection protocol module 1110. The conventional Internet browser provides an Internet gateway to the destination computer 108A. The client uses the Internet browser 1102 to request access to the web site stored in the server 10 storage module 208. The menu files 1108 include the files received from the server 102 in response to the access request. In addition, the personal computer storage module 1100 receives the Vayu Web client module 1104 and the web page preload module 1106 from the server 102. The operation of these modules is described in greater detail below.

In one embodiment, the Vayu Web client module has been implemented using three 15 different techniques. These three techniques allow the client module to operate across the wide variety of client machines currently in use across the Internet, e.g., IBM PC, Apple Macintosh, Sun Microsystems Sparcstation, and across the wide variety of WWW Browsers currently in use, e.g., Netscape versions 2.n and 3.n, and Microsoft Internet Explorer versions 2.n and 3.n. The three techniques are: (1) as a Java Applet; (2) as a Javascript script where Javascript is a 20 Internet language developed by Netscape Corporation; (3) as a Client-Pull mechanism using a <META> tag in the HTML file, that causes the browser itself to automatically load pages on a timed basis. The CONTENT argument contains the time in seconds. The URL contains the target URL. The <META> tag can be:

```
<META HTTP-EQUIV=REFRESH CONTENT="1 URL=target_url">
```

25 Figure 12 is a flow chart describing the method performed by the Vayu Web client module according to one embodiment of the present invention. As described above, the client requests information from the web site server 102 and the destination computer 108A downloads 1202 web site server information. The web site server information initially downloaded includes the Vayu Web client module 1104, the web page preload module 1106, 30 and the new site project data structure that is stored in the menu files 1108. As described above, the new site project data includes information about the web site and includes references to web pages in the web site but does not include all of the web page data. The destination computer 108A implements 1204 a web page preload routine that is executed concurrently with the

remaining functions performed by the Vayu Web client module, as described below. Figure 13 is a flow chart describing the method performed by the web page preload module 1106 according to one embodiment of the present invention. The web page preload module 1106 gathers local environmental information concerning the client personal computer. This information includes the speed of the modem, the time of day, Internet service provider service level, browser type and version, and operating system type and version. This information is used by the artificial intelligence expert system to dynamically select a preload strategy to optimize the performance of the presentation of web pages to the client. The web page preload module 1106 preloads 1304 the first web page from each of the menus.

10 While the web page preload module 1106 is preloading the first web page associated with each of the menus, the Vayu Web client module 1104 displays 1206 the menu page and waits for the user to select a menu. When the user selects 1208 a menu item, the Vayu Web client module 1104 identifies 1210 the first web page in the predefined sequence of web pages associated with the menu item and displays 1212 the first web page. Before the user selects a menu item, the web page preload module 1106 preloads the first web page associated with each menu item, as stated above, and then determines the web page loading sequence for subsequent web pages based upon a determination of the most likely menu item selection. The web page preload module 1106 determines the most likely menu item selection based upon the selections of previous clients or, if no clients have previously accessed the web page, the webmaster may indicate which menu item to show first, or the first menu item is selected as the most likely to be selected. The web page preload module 1106 preloads web pages from the sequence of web pages associated with the most likely menu item selection by implementing the rule based expert system of the present invention, described below.

25 After the client selects a menu item, the Vayu Web client module 1104 and the web page preload module 1106 receive the web page sequence and web page information from the new site data structure in the menu files 1108. Specifically, the new site data structure includes the overall size of the web page, the number of embedded objects in each web page, and the size of the embedded objects in each web page, as described above. In one embodiment of the present invention, the web page preload module 1106 determines a priority for each web page in the web page sequence based upon the environmental parameters of the destination computer 108A, described above, a queue location of the web page in the predefined sequence, the size of the web page, the number of embedded objects in the web page, and the size of the embedded objects in the web page, for example. In addition, the web page preload module is keeping track

of how long it actually takes to load each page. From this information it calculates the actual throughput that the current client is getting. This throughput is used by the expert system to make decisions about preload strategies. For example, if the throughput falls below a certain value, the preload module may stop preloading certain types of MIME files (e.g., video, sound).

- 5 The trigger value would be set by the web master during the menu configuration stage, described above. One example of a rule based expert system is set forth in Table 2.

```
10 ; Set preload strategy based upon throughput
    IF
        system.throughput < 6kpbs
    THEN
        preload.order = STRATEGY_7
    ENDIF

15 ; Set preload strategy based upon throughput
    IF
        system.throughput < 2kpbs
    THEN
        mime.image = NOLOAD
    ENDIF

20 ; Set priority based upon size of an embedded object
    IF
        object1.size > 250000
    THEN
        object1.preload_priority = 5
    ENDIF

25 ; Set priority based upon size of an embedded object
    IF
        object1.size > 500000
    THEN
        object1.preload_priority = 7
    ENDIF

30 ; Set priority based upon size of an embedded object
    IF
        object1.size > 1000000
    THEN
        object1.preload_priority = 10
    ENDIF

35 ; Set priority based upon the number of embedded
    ; objects in a page
    IF
        page1.objects > 5
    THEN
        page1.preload_priority = 3
    ENDIF

40 ; Set priority based upon the number of embedded
    ; objects in a page
    IF
```

```

        pagel.objects > 10
    THEN
        pagel.preload_priority = 8
    ENDIF
5
    ; Set preload order based upon browser type and time
    of
    ; day
    ; MSIE_3x is Microsoft's Internet Explorer Version 3.x
10    IF
        ; night time hours
        browser.type = MSIE_3x AND (env.timeofday > 20:00
    AND env.timeofday < 08:00)
    THEN
15        preload.order = SEQUENTIAL
    ENDIF

    ; Set preload order based upon browser type and time
    of
20    ; day
    ; STRATEGY_7 is load page 1, then 2, then largest to
    ; smallest
    IF
        ; business hours
25        browser.type = MSIE_3x AND (env.timeofday >
    08:00 AND env.timeofday < 20:00)
    THEN
        preload.order = STRATEGY_7
30    ENDIF

```

Table 2

After determining the preloading priority for each web page, the web page preload module 1106 preloads 1308 the web pages having the highest priority. The web page preload module 1106 monitors 1310 the preloading procedure to reduce any delay in displaying web page to the client. The monitoring function of the web page preload module 1106 permits the preload function to be dynamic in that if a problem occurs 1312 in the web page loading process, the web page preload module 1106 sorts and reorganizes 1314 the data. Examples of problems include: (a) when a page is needed to be displayed and has not yet been preloaded or

40 (b) when the preloading is sufficiently ahead of the display so that the local PC web page cache is full, and pages must be discarded (and reloaded). When a problem occurs the expert system is re-initialized with the current data (including the throughput and modified menu files without the web pages that have already been displayed). A modified preload strategy is then determined 1306.

45 The client continues navigating 1210 through web pages while the destination computer 108A continues displaying 1212 web pages. As described above, in one embodiment of the present invention, the client can passively watch the sequence of web page displays. In alternate

embodiments, the user has the option of selecting a next page button. Within each web page the client also has the option of manipulating a control panel. Figure 14 is an illustration of a web page control panel 1400 according to one embodiment of the present invention. In this embodiment, the user has the option of returning to the beginning of the web page by selecting control panel button 1402, proceeding to the end of the web page (for example, a video clip embedded in the web page may be skipped) by selecting control panel button 1404, continuing or beginning the execution of the web page display by selecting control panel button 1406, pausing the execution of the web page by selecting control panel button 1408, decreasing the speed of web page execution by selecting control panel button 1410, increasing the speed of web page execution by selecting control panel button 1412, or stopping web page execution by selecting control panel button 1414.

An alternate (preferred) embodiment of the present invention is described below with reference to Figures 15-22. This alternate embodiment includes a standard structure and navigation for the menus (pages containing buttons) and menu items (buttons), allowing the user a set of options at each web page. One benefit of using the standard structure is to make the web site easier to use by the end user and to offer the web site designer more control over what is displayed to the user.

Figure 15 is an illustration of a first menu web page according to an alternate embodiment of the present invention. The first menu page 1500 illustrated in Figure 15 includes a single large button 1502 that includes the name of the company that is the subject of the advertising, e.g., "Vayu Web". The user can select the button 1502 to continue or the user can exit from the web site. In addition, the present invention enables the company to add a graphic display to the button or to the background to enhance the screen display.

The first menu web page can include a tool bar 1504 having various control buttons thereon. These control buttons are enabled and disabled for each web page based upon the web site definitions defined by the webmaster. When the user selects the button 1502, a sequence of pages may automatically be presented after which another menu page appears containing more items. Alternatively, the sequence can be skipped and the next menu page can be presented immediately. While the user is viewing a dynamic sequence, the control functions are typically disabled. The control functions include pause, fast forward, reverse, and exit. In alternate embodiments these control functions are enabled during the presentation of dynamic sequences. While a dynamic sequence is playing the user typically watches the display until the next menu page is reached.

Figure 16 is an illustration of a second web site menu page according to an alternate embodiment of the present invention. At the next web site menu page 1600, the user has three options. (1) return to the previous menu page, (2) select one of the menu items 1602, or (3) select the auto-play button 1604 to automatically cycle through the sequences following each menu item, for example. The order of cycling through the menu items is determined by the web site designer (web site master). Some items can be excluded from this "autoplay". A menu page can comprise any number of items. In this alternate embodiment the number of items is typically between 1 and 10. The web site designer can add their own graphics to the items or to the page background.

10 Certain web pages can be static, i.e., they do not automatically display another web page. Frequently, for static web pages, the "pause" and "forward" buttons on the tool bar 1504 are enabled. In this case, the page may include a description of a product and/or ordering information, for example. A maximum viewing time can be specified by the web site designer to proceed to the next page after some amount of time has elapsed, or the web site designer can permit the user to continue to the next web page by selecting a next page control button.

15 In this alternate embodiment, the tool bar 1504 is located at the top of the screen. The web site designer may choose not to display the tool bar 1504 for certain screens when, for example, the user has no other options within these screens but to wait for the next page. The tool bar 1504 can also include status indicators 1606 to show the remaining time left to view this page. This tool bar 1504 can be used for every page displayed using the present invention.

20 Another feature of the present invention is that the user is permitted to create multiple versions of each web site where the present invention permits optimization of each web site for different internet throughput rates. For example, the throughput range for different types of Internet access devices commonly used today varies from 14.4 kbps (modem) to 56 kbps (ISDN) to 1.5 megabits/sec (T1). This variation exceeds a factor of 100. Furthermore, users of the present invention on Intranets, e.g., local area networks (LANs) can reach data transfer rates exceeding 10 megabits/sec. Due to transfer rate constraints at the lower throughput speeds, the web site designer may include several versions of a web page, or several versions of a sequence, where a sequence includes a series of pages which are automatically displayed, with each version being a different size to optimize the web site display for different throughput environments. The database can include information about the various page and sequence versions. The artificial intelligence module of the present invention that was described above with respect to the expert system and is described in greater detail below, can automatically

select the appropriate page version depending on the throughput environment it has determined for a user. The Virtual Design Studio, which is the tool that the web site designer uses to assemble the web pages into advertisement sequences, is designed to assist the web designer in creating optimal sequences for various throughput environments. The Virtual Design Studio
5 will also allow the web site designers to preview the web pages of the web site using a simulator of the present invention in order to simulate environments having various data throughput rates.

It will be apparent to persons skilled in the art that the use of different versions of the web pages is not required to operate the present invention. However, if the web site designer is expecting users with different throughput environments to access the web site, it may be
10 advantageous to create versions of certain web pages that contain videos, sound, etc., so users with faster connections can view web pages having greater detail, while users with lower connections can still get the full benefits of this web site by viewing pages of a smaller size.

The present invention permits different versions of web pages and web sites by providing the web site designer with the option of providing high-quality (larger) pages for
15 users connected at higher throughput rates and lower-quality (smaller) pages for users connected at lower throughput rates. The present invention enables the web site designer to easily mix the smaller pages with the larger pages when the throughput is fast. However, adding larger pages to a low-throughput connection requires the designer to carefully sequence the pages so a large page is preloaded while several smaller ones are being viewed. In addition, some pages take
20 longer to view, for example if they contain detailed textual information that takes longer to read. This longer viewing time can be used to preload larger files. The Virtual Design Studio of the present invention includes modules which assist in the intelligent sequencing of pages in this manner not only to reduce the user's waiting time but also to improve the quality of the pages that can be viewed with slower connections.

Another factor to consider is that the actual throughput rate during an Internet
25 connection session can vary greatly. For example, with a nominal connection speed of 28.8 kbps, the actual transfer rate can easily vary between 10 kbps and 28.8 kbps. Therefore, the artificial intelligence (AI) module of the present invention makes real-time adjustments during a connection session to keep the presentation flowing and prevent waiting time for the user. One
30 of the techniques implemented by the AI module is to leave out certain pages that the designer has identified as being large but with low impact or information content. Another technique is to dynamically switch a higher-quality presentation to a lower one when the throughput slows

down, and then to switch back to the higher-quality one if the throughput picks up again. The artificial intelligence unit is described in greater detail below.

Figures 6, 8, 9, and 10 are illustrations of screen displays during the analyze web site process 408 and the configure web site process 410 for one embodiment of the present invention as described above. Figures 17-22 are illustrations of screen displays displayed to the web site designer during the analyze web site process 408 and the configure web site process 410 according to an alternate embodiment of the present invention. The Virtual Design Studio is stored in the web site analyzer 310 and enables a web site designer to: (1) construct presentation sequences from pre-existing and newly created web pages; (2) optimize the presentation sequences for different throughput environments and specify alternatives to be used by the artificial intelligence module to compensate for throughput variations in real time; and (3) create menu structures and user navigation through the web site. The virtual design studio includes a site repository information screen that is illustrated in Figure 17. The site repository information screen allows a web site designer to view, insert, delete and edit web page information and web page sequences. For example, the site repository information screen allows the web site designer to assemble pages into the sequences which form the basis of the automated web site content. The site repository information screen also enables the web site designer to modify and maintain the web site.

Figure 18 is an illustration of a screen display of the web site presentation properties of various web site sequences developed by the web site designer. Figure 18 illustrates two alternate web sites, the first web site 1802 is defined to be accessed by clients having a high data transmission connection, e.g., a client having a T1 line. The second web site 1804 is defined to be access by clients having a low data transmission connection, e.g., a client having a 14.4 or a 28.8 kbps modem connection.

Figure 19 is an illustration of a web page sequence properties screen display 1900. The web page sequence properties screen display 1900 identifies the optimal data connection rate for a particular web site sequence. In Figure 19, the optimal user has a connection rate between 14.4 kbps and 33.3 kbps. The web page sequence properties screen display also enables the user to identify a slower alternate web page sequence and a faster alternate web page sequence that can be selected by the artificial intelligence unit, as described below.

Figure 20 is an illustration of a web page property screen display according to an embodiment of the present invention. The web page property screen display 2000 includes information about various components of each web page including the size and download time

of each component, whether the web page is static or dynamic, as defined above, and defines various timing features of the web page including the nominal load time, the ideal viewing time, the minimum viewing time and the maximum viewing time. In addition the page property enables the user to select various options including the display of control buttons, e.g., the forward and pause button, and enables the user to identify web pages that can be skipped if the actual data throughput is slower than the anticipated data throughput, as determined by the artificial intelligence unit, described below.

Figure 21 is an illustration of a sequence optimizer screen display according to an alternate embodiment of the present invention. The sequence optimizer screen display allows the user to optimize the arrangement of web pages within a web site and to optimize the length of time each web page is displayed. The sequence optimizer screen display is an easy-to-use graphical interface that displays the ideal and estimated viewing times for each page. The sequence optimizer is part of the web site configuration module 312 illustrated in Figure 3. The sequence optimizer of the present invention calls a web page display simulator that is also part of the web site configuration module 312 which provides feedback to the web site designer regarding the presentation of the sequence of web pages to the user under different network throughput conditions. The present invention determines the ideal and estimated viewing times based upon the amount of text in the web page, the level of detail in visual objects, and the length of video objects, for example. The user can modify the viewing time for each page.

Using this ideal viewing time and the estimated page loading time under various network throughput conditions, the Sequence Optimizer graphically displays for each page the amount of credit time, i.e., the time differential between the viewing time and the time at which the page has been preloaded, or the amount of debit time, the time differential between the time at which the page will be preloaded and the viewing time. The user can adjust the parameters for each page and see this credit/debit value change for each page. The parameters that can be set for each page include: the ideal view time, which is the amount of time the designer wants page to display, and the optional view flag, which indicates whether a page can be skipped over if the system is too busy, for example.

In addition the Virtual Design Studio unit automatically computes for each page the estimated minimum view time, which is the minimum time estimate that the page must be displayed before the next page is ready for display, and the normal load time, which is an estimate of the amount of time it will take to download a page in the throughput environment that the designer has set for this sequence.

The site layout studio unit is part of the web site configuration module 312 and enables the web site designer to create menus and define the web page sequence of the web site. The site layout studio unit displays a site menu page display and a menu item that displays the name of the site. When a user selects the menu item, one of at least two things happen: (1) a sequence
5 of pages will play, or (2) another menu will appear, each of which can contain up to ten menu items. Using this model, sequences and menus can be nested infinitely. Figure 22 is an illustration of the site layout screen display that is displayed to the web site designer according to an alternate embodiment of the present invention. The site layout screen display 2200 includes a menu portion 2202, a menu layout portion 2204, and a control portion 2206. The
10 menu portion 2202 includes the tree of menus and menu items within a site and the links there between. Selecting an object expands it and shows the menus and items linked hierarchically below it. The menu layout portion 2204 displays the screen layout of the menu items on each menu. Selecting a menu in the upper screen causes that menu to be displayed in the menu layout portion 2204. The user has the option of changing the menu style, by selecting the menu style button 2208 in the control portion 2206 in terms of the types of standard menu styles
15 supported by the present invention, e.g., by modifying the displayed graphics and colors.

The virtual design studio unit includes a relational database that resides on the same computer the Virtual Design Studio is on, which can be the web site designer's computer. This database contains all the information regarding the automated web site except for the pages
20 themselves, which are stored in files on the web server. A conventional HTML Generator module of the Virtual Design Studio creates an encoding of the database contents into a format that can be easily read by the client-side software on the user's computer. When the web site designer has completed defining the menus, pages, and sequences for the automated web site, the web site designer runs the HTML generator to create a file which is placed on the web server
25 computer. These generated files contain the HTML code which the client-side processor and AI processing module uses to perform.

The web page preload module 1106 is described above with reference to Figures 11-13. In an alternate embodiment the web page preload module includes an artificial intelligence (AI) module that performs the functions described below. The artificial intelligence module supports
30 the selection of different page versions and additional real-time decisions on page preloading. In other embodiments the artificial intelligence module can be stored on the server instead of the client computer.

After a user connects to the server having the web site, the AI module is downloaded to the client computer and collects information about the client computer environment including the operating system, processor capacity, Java support capabilities, types of video and audio formats that are supported, and the current throughput rate of the Internet connection. Using this information the AI module makes decisions regarding which version of Java to use and which version of the page sequence to use. Since network conditions can change during the session, the AI module continually monitors the network conditions, e.g., the rate of data throughput, and can modify the web page sequence by skipping a web page, for example, in order to reduce the waiting time for the user. The AI module periodically assesses the state of the session, and makes decisions accordingly. The functions performed by the AI module includes: (1) preloading pages; (2) canceling the preloading process; (3) modifying the rate of play of the current sequence; (4) inserting an additional page, e.g., a message, into the sequence; (5) skipping a page in a sequence being presented; (6) replaying one of the components, e.g., sound or video; and (7) switching to presenting another sequence of pages with a different ideal throughput requirement. A more detailed description of the AI module operation is described below.

In the alternate embodiment the AI module is downloaded to the client computer. Specifically, when the user contacts the web site, a client module is downloaded on the first page and the HTML code output by the HTML generator module of the virtual design studio, described above is also downloaded. This HTML code is used by the client code module and the AI module, and contains the complete description of the web site in terms of the menus, menu items, sequences, and pages found in each sequence. The client module performs three major functions: (1) probing the client machine to determine the client environment and the throughput rate, and uploading the HTML code; (2) displaying menus and pages; and (3) processing user toolbar input.

The client module implements the site template as designed by the web designer. The first menu page is displayed, and the user can choose one of the available options. When the user selects an option, the sequence of web pages associated with that option is displayed in sequence on a display device, e.g., a computer monitor, coupled to the client computer. Each web page can be displayed for a certain period of time, and the client module is responsible for switching pages after the view time has expired.

As stated above, some web pages may have an active toolbar. For example: some pages with a great deal of text may require that the user indicate when the user has finished reading the

text. It is possible to have a single view time that would work for all users, however fast readers may become frustrated if the view time is too long since the view time would need to be long enough to accommodate slower readers.

The client-server interaction and processing on the Internet World Wide Web is
5 complex, and is greatly affected by factors such as network traffic, the number of clients
accessing a particular site, the amount of content the client requested, the computer hardware
speed, and the software efficiency. While some of these factors (e.g. modem speed) stay the
same during an Internet session, other ones, such as network traffic, varies constantly.

To deliver an uninterrupted or nearly uninterrupted presentation to the user, the AI
10 module frequently monitors the network, the server, and the client environment during a
session, and makes complex decisions regarding how to proceed with the presentation. The AI
module accounts for processes happening both on the server and on the client computer. The AI
module synchronizes these processes while gathering information and while making decisions.
The AI module constantly performs "what-if" scenarios, examining different options for
15 preventing interruptions to the pages appearing on the user's display screen.

The AI module relies on the sequences being constructed in an optimal manner for
nominal throughput for a particular environment. For example, if the nominal throughput for
the sequence is 20 kbps, the AI module presumes that the sequence was constructed to flow
smoothly as long as the connection uniformly maintains this throughput. However, the present
20 invention is designed to work under various network conditions, under different server
responsiveness, and with a variety of client computer environments, e.g., low-end personal
computers with basic web browsers or high-end workstations supporting more powerful web
browsers.

As described above, at the beginning of the client session, the AI module determines the
25 client's environment and the current state of the Internet connection. Some of the parameters
which will affect the decisions of the AI module are: (1) the processing and Input/Output (I/O)
speed of the server machine; (2) the throughput capabilities of the Web Server software; (3) the
operating system on the client machine; (4) the client computer's support of Java applets; (5) the
client computer's support of JavaScript; (6) the client computer's support of Client Pull
30 technology; (7) the types of video and audio formats that can be played by the browser or plug-
in components of the client computer; (8) the maximum number of simultaneous connections to
the Server that the client permits; (9) the browser cache size of the client computer; (10) the

number of users currently communicating with the Web Server; and (11) the current throughput rate of the Internet connection.

In one embodiment of the present invention, the AI module determines the following parameters at the beginning of a session and monitors the following parameters throughout the session: (1) the number of users currently communicating with the Web Server; (2) the current
5 throughput rate of the Internet connection; and (3) the volatility of the network throughput rate.

At the beginning of each presentation, the AI module, based on the parameters of the session: (1) selects the versions of the requested presentation that are supported by the client environment, e.g., Java, JavaScript, ClientPull, or Generic; and (2) selects the starting version of
10 the presentation. As described above, since network conditions may change during the presentation, the present invention can utilize all of the selected versions by, for example, switching between web page sequence versions in response to a variations in network conditions. In addition, the present invention sets the following parameters: (1) the maximum
15 number of simultaneous connections the present invention utilizes; (2) the rate of change of the speed of the presentation sequence in response to change in network conditions; and (3) the types of filler pages, filler applications and special effects to be used during the presentation. In addition, as the session progresses, the AI module preloads new pages, graphics, sound, video and other presentation components. The AI module uses the information obtained during its
20 network monitoring operation to determine whether the original sequence still can be played without interruptions. The AI module also monitors the progress of each component being preloaded from the Web Server, in order to make decisions about when to cancel preloading, or initiate new component preloading, as described above.

Appendix A includes a psuedo-code representation used to implement the AI module and is incorporated by reference herein.

25 The following is an example of the operation of the AI module. After the client connects to a server, the present invention determines, for example, that the client computer utilizes the Windows 95 Operating System, supports Java applets, supports running QuickTime videos, permits a maximum of two simultaneous connections to the server, and has a browser cache size of 10 megabytes. The AI module determines that versions of the top-level presentation can
30 utilize Java applets and QuickTime video technology.

A probe page is downloaded to the client computer, and the present invention determines that the number of users currently communicating with the web server is at about half of Server's capacity and the current throughput rate of this Internet connection is 1000 kbps.

The present invention then selects the starting version of the presentation that corresponds to network throughput rate of 1000 kbps, and preloads the first page of the web site. The AI module determines that there are two components currently being downloaded, one from the next page, and one from the following page.

5 The presentation begins, and the present invention starts presenting selected sequence of pages, while continuing to monitor the presentation and network conditions.

 During the presentation, and as a result of the network monitoring performed by the AI module, the present invention detects a drop in network connection's throughput, for example down to 100 kbps.

10 The AI module of the present invention sets up a virtual scenario of canceling the process of preloading the second component because even though the first component would download sooner, the timing for the next page would still be late.

 The AI module determines that current page has an animated GIF component that can be replayed at the end of the page and adds this replay to the scenario being considered. However,
15 invoking a what-if analysis still results in unacceptable timing. Therefore, the expert system consults a presentation script, and determines that at this point a "Skip page", or "Special Effect" actions are not available and, accordingly, rejects this scenario.

 The expert system then runs through some other viable "what-if" scenarios, and rejects
20 all of them except the one requiring switching to the sequence corresponding to a 56 kbps throughput.

 The AI module then instructs the Vayu Web client module 1104 to modify the selected sequence of pages. The AI module then continues monitoring the presentation and network conditions.

25 Later in the presentation, the AI module detects an increase in network connection's throughput up to 900 kbps. The expert system sets up a virtual scenario of switching to the sequence corresponding to 1000 kbps throughput, and runs a what-if analysis on that scenario, whose result is acceptable. The AI module then switches back to the sequence of pages corresponding to 1000 kbps throughput, and continues with the presentation.

30 While the present invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by persons skilled in the relevant art that various changes in form and details can be made therein without departing from the spirit and scope of the invention.

Appendix A

5 The following is Java-based pseudo-code for the AI module
for an embodiment of the present invention.

Class Definitions

```

10 class PrSite
    //describes the site (e.g. "General Goods")
    //for which presentations are built
    string name;
    PrSiteViewVec & siteViewVec; //presentation projects
15 };

class PrSiteView
20 {
    //hierarchical tree of presentations accessible to user logging-in
    string name;
    PrSite & site;
    PrPage & rootMenuPage; //top-level menu
25 Presentation & rootPresentation; //top-level presentation
};

typedef Vector<Presentation> PresVec

30 class Presentation
{
    //describes a presentation corresponding to a menu item
    string name;
    string type; //sequence, white-paper, order-form
35 PrSiteView & siteView;
    Presentation & parent; //parent in the pres. tree
    PresVec & childVec; //first child in the pres. tree
    SequenceVec & sequenceVec; //list of alternate page-sequences
    PrPage & tailMenuPage; //menu to display after presentation
40 PrPageVec & tailPageVec; //list of pages tacked-on (e.g. order-
    //form)
};

45 class PrSequence
{
    //sequence of pages making up a presentation
    //a presentation may have several alternative sequences
    //for different communication speed, Java support, plug-in support
50 string name;
    string type;
    string environmentType; //e.g. for Java-supporting browsers
    Presentation & presentation; //back pointer to presentation owning
    PrSeqItemVec & itemVec; //list of items (pages) comprising this
55 };

class PrSequenceItem
{
    //item of the presentation sequence
    PrSequence & sequence;
60 PrPage & page; //page (usually HTML)
    long idealViewTime; //prescribed under normal state
    long minViewTime;
    long maxViewTime;
    long loadTime; //expected load time at throughput
65 //for which sequence was designed

```

-28-

SUBSTITUTE SHEET (RULE 26)


```

    int      pauseAllowed; // user pauses when this item is displayed
    int      forwardAllowed; // enable "forward" button when displayed
    int      skipAllowed; // allow skipping this page
5   PrPage & preloadPage; //when displaying this item, start preloading
    //preloadPage before preloading next page
    PrFxVec & fxVec; //list of available "special effects"
    //to be used when timing is Late
    PrSequence & upSequence; //another sequence to switch to
    //when Internet conditions improve
10   PrSequenceItem & upItem; //item within sequence to switch to
    //when Internet conditions improve
    PrSequence & downSequence; //another sequence to switch to
    //when Internet conditions degrade
15   PrSequenceItem & downItem; //item within sequence to switch to
    //when Internet conditions degrade
};

class PrPage
20 {
    //HTML page, or special effect page, or applet
    string      name;
    string      type; //HTML, VRML, Applet, FX
    string      url; //locator
25   long      size; //file size (for download monitoring)
    PrComponentVec & componentVec; //list of components (images, sound, video)
};

class PrComponent
30 {
    //image, video, sound, applet and other non-text components of a page
    string      name;
    string      mime_type; //image, sound, etc.
    long      size; //file size
35   long      playTime; //play (display) time; for timing
    string      url; //locator
};

class PrFx
40 {
    //special effect (filler to smooth out the presentation at changing conditions)
    string      name;
    string      type; //text msg, applet, animation clip
    string      selectionType; //random, by priority, specific
45   PrPageVec & pageVec; //page containing this FX
};

class PrEnvironment
50 {
    //class of browser environment
    //for which sequences can be created
    string      name;
55 };

#include "VauyWebPresSmooth.h"

60 class PresentationManager : public Applet
    {
        int ProbePage()
        {
65             //spawn a separate thread to gauge throughput rate
            spawnThread( "ThroughputTest" );
        }
    }

```

```

5 //get maximum number of simultaneous connections
  //the browser is allowed to open
  maxConnectionCnt = Browser.getMaxConnectionCnt();

  //get browser's buffer size
  netBufferSize = Browser.getNetBufferSize();

10 //inquire what kinds of plugins the browser has
  Browser.getPlugins(pluginVec);
  for( plugin = pluginVec.firstElement();
      plugin != NULL;
      plugin = plugin.nextElement())

15     switch plugin.type
    {
      case pluginStreamingVideo:
        streamVideoType = StreamVideoTypeVec(plugin.vendor);
        break;
20     case pluginVirtualReality:
        vrPlayer = Yes;
        break;
    } //switch

25 // wait for completion of the throughput test thread
  waitFor( finish, "ThroughputTest");

30 //now we have all parameters needed to determine
  //which of the presentation environment to choose
  environ = determineEnvironment();

35 //preload the top-level menu, and display it
  spawnThread( "PreloadPage", presView.rootMenuPage);
  curPage = presView.rootMenuPage;
  displayPage( curPage);

40 //determine which sequences will potentially be used
  //in the presentation, as well as starting sequence
  rootPresentation.startSequence = DetermineSequences( sequenceVec);

  //preload first page of the chosen sequence
45  spawnThread( "PreloadPage", "Fx", presView.rootPresentation);

  //spawn controlling thread
  spawnThread( "rootSequenceManager");

50 //spawn a separate thread to monitor throughput rate
  spawnThread( "PeriodicThroughputTest");

  //ProbePage()
}

55 int Presentation::SequenceManager()
{
  curSequence = startSequence;
  curEnv = startEnv;

60 //go through all items of the sequence
  seqItem = mainSequence.pageVec.firstElement;
  while( true)

65     //reassess the state of the network (since prev. page)

```

```

curState = assessState();
switch curState
{
5   case stThroughputDidNotChangeDramatically:
      //can we preload one more component?
      if( preloadedCount >= MaxPreloadedCount)
      {
          break;
10      }
      nextPreloadSeqItem =
          curSequence.getNextPreloadSeqItem();
      nextPreloadComp =
          curSequence.getNextPreloadComponent();
15      //check if this comp is already cached
      nextPreloadComp.calcDownloadTime( curEnv);
      nextDownloadOK = true;

      //for each comp being downloaded,
      //see if starting new download will thwart timing
20      for(i = 0; i < MaxPreloadedCount; ++i)
      {
          comp = componentBeingDownloaded( i);
          if( comp.status != stInProgress)
              continue;
25          comp.calcTimeToStart();
          comp.calcRemainingTimeToDownload();

          if( comp.timeToStart <
              comp.remainTimeDL +
30              nextPreloadComp.totTimeDL)
          {
              //timing = Late; it is not OK to download
              nextDownloadOK = false;
              break;
35          }
      }
      if( !nextDownloadOK)
          break; //switch
      //OK to download this component; do it
40      spawnThread( nextPreloadComp.preload);
      break; //switch

      case stUrlNotFound: //file not found
      //can we skip this page?
45      if( nextPreloadSeqItem.skipAllowed)
          //yes: skip it
          nextPreloadSeqItem.setSkip();
          else
          //no: try to switch sequences
50          nextPreloadSeqItem.setJumpSequence();

      case stThroughputIncreased:
      //keep count of how steady is the increase
      ThroughputIncreasedCount++;
55      ThroughputDecreasedCount = 0;
      nextSequence.nextItem = curSequenceItem.upSeqItem();
      //can we afford (timing-wise) to go up the sequence
      nextSequence.nextItem.calcTiming();
      if( nextSequence.nextItem.timingOK)
      //yes: switch
60      setNextItem( nextSequence.nextItem);
          else
          //no: act as if no change
65      setStatus( stNormal);

```

```

break;

case stThroughputDecreased:
    //keep count of how steady is the decrease
    ThroughputDecreasedCount++;
    ThroughputIncreasedCount = 0;

    calcCurTiming();
    if(curSequenceItem.timingOK)
        //we are still OK for this and next page
        break;

    //try different ways to handle oncoming delay.
    //(returns success if we can handle it)
    if((status = handleLateTiming()) == Success)
        break;

    //cannot handle delay with any other method
    //try to switch to lower sequence
    curSequence = curItem.lowSequence;
    nextItem = curItem.getLowSeqItem();

    break;

case stThroughputLow:
    //keep count of how steady is the decrease
    ThroughputDecreasedCount++;
    ThroughputIncreasedCount = 0;
    calcCurTiming();
    if(curSequenceItem.timingOK)
        //we are still OK for this and next page
        break;
    else
    {
        setStatus(stThroughputDecreased);
        continue; //next iteration picks it up and handles
    }
} //switch
} //while
} //manager()

int handleLateTiming()
{
    //if we currently are displaying text,
    //increase viewing time by 20%
    //(whether it helps or not)
    if(curItem.page.mimeType = "text")
    {
        increaseViewTime(curItem, percentIncreaseForText);
        calcCurTiming();
        if(curSequenceItem.timingOK)
        {
            return Success;
        }
    }
}

//can we replay any components already loaded?
if( curItem.replayCompVec != NULL)
{
    //rule: if comp i is replayed,
    //then i+1; i+2; ... must be replayed
    comp = curItem.replayCompVec.lastElement();
}

```

```

5         while (comp != NULL)
            virtualAddComp( comp);
            calcCurTiming();
            if( curSequenceItem.timingOK)
            {
                curItem.setReplay( comp);
                return Success;
            }
10         else
            //try one more replay-components
            comp = comp.PreviousElement();
        }
15     //are we preloading two pages ahead?
    //see if stopping that will help
    //if it does not help, do NOT stop preloading it yet!
    while(1)
    {
20         seqItem = getLatestItemForWhichCompBeingPreloaded();
        if( seqItem != curItem
            && seqItem != curItem.nextElement()
        )
        {
            virtualStopDownload(seqItem.downloadingComponents);
            calcCurTiming();
25         if( curSequenceItem.timingOK)
            {
                //we will be OK for this and next page
                stopDownload(seqItem.downloadingComponents);
                return Success;
30         }
        } else
            break;
    } //while
35     //can we skip next page (or couple)?
    while( curItem.nextElement.skipAllowed)
    {
        curItem.nextElement.setSkip();
        calcCurTiming();
40         if( curSequenceItem.timingOK)
            {
                //we will be OK for this and next page
                return Success;
            }
    }
45     //try FXs available
    if( curItem.FxVec != NULL)
    {
50         fx = curItem.FxVec.firstElement();
        while( fx != NULL)
        {
            virtualApplyFx( fx);
            calcCurTiming();
            if( curSequenceItem.timingOK)
55         {
                //we will be OK for this and next page
                return Success;
            }
        }
    }
60     return Fail;
} //handleLateTiming()
} //class

```

CLAIMS

What is claimed is:

- 5 1. A computer based method for generating a web site having a plurality of web pages, the computer having a storage module and having a wide area network interface, the method comprising the steps of:
- designing a menu web page having one or more menu items:
- designing a sequence of web pages, including a first and second web page, for each of
- 10 said menu items, said first web page only capable of accessing one of said second web page and said menu web page, said second web page only capable of accessing the immediately subsequent web page in said sequence of web pages and said menu web page, said sequence of web pages:
- identifying each menu item with one of said sequence of web pages: and
- 15 storing said menu web page and web page sequence information in the storage module.
2. The method of claim 1, wherein said sequence of web pages is designed by a webmaster, said sequence of web pages enables the webmaster to control the content and
- 20 sequence of information displayed to the client.
3. The method of claim 2, further comprising the step of:
- receiving a request signal from a destination computer via the wide area network interface, for requesting access to the web site:
- transmitting said menu web page and said web page sequence information to said
- 25 destination computer via the wide area network:
- displaying said menu web page to a client on a display device coupled to said destination computer:
- preloading additional web pages before said client selects a menu item:
- receiving a menu item selection from a client:
- 30 preloading said sequence of web pages associated with said selected menu item, using a dynamic preload algorithm: and
- displaying said sequence of web pages to the client without permitting the client to alter said sequence.

-34-

SUBSTITUTE SHEET (RULE 26)

4. The method of claim 3, wherein said step of preloading said sequence of web pages includes the steps of:

determining a data receiving rate of the destination computer;

5 determining the size of two or more web pages in said sequence of web pages:

setting a priority value for each of said two or more web pages in said sequence of web pages based upon said data receiving rate, said size of said web page, and the position in said sequence of said web page;

10 preloading one of said two or more web pages into said destination computer storage module, based upon said priority value; and

dynamically modifying said priority value of one or more web pages to reduce any delay in displaying one of said web pages to the client.

1/17

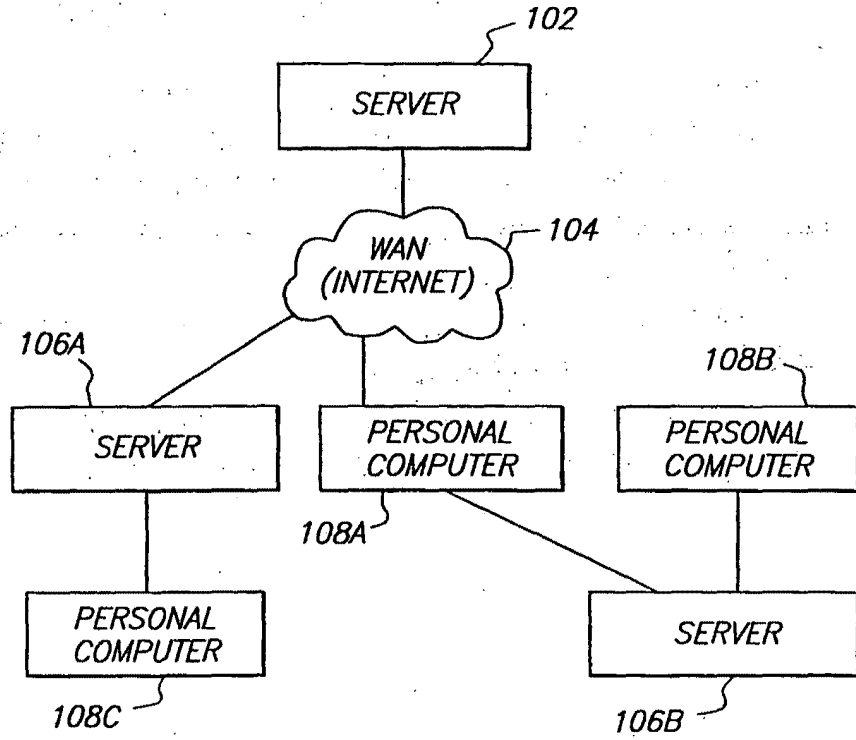


FIG. 1

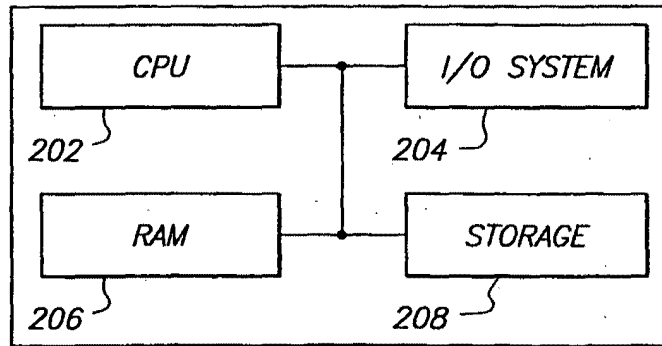


FIG. 2

SUBSTITUTE SHEET (RULE 26)

2/17

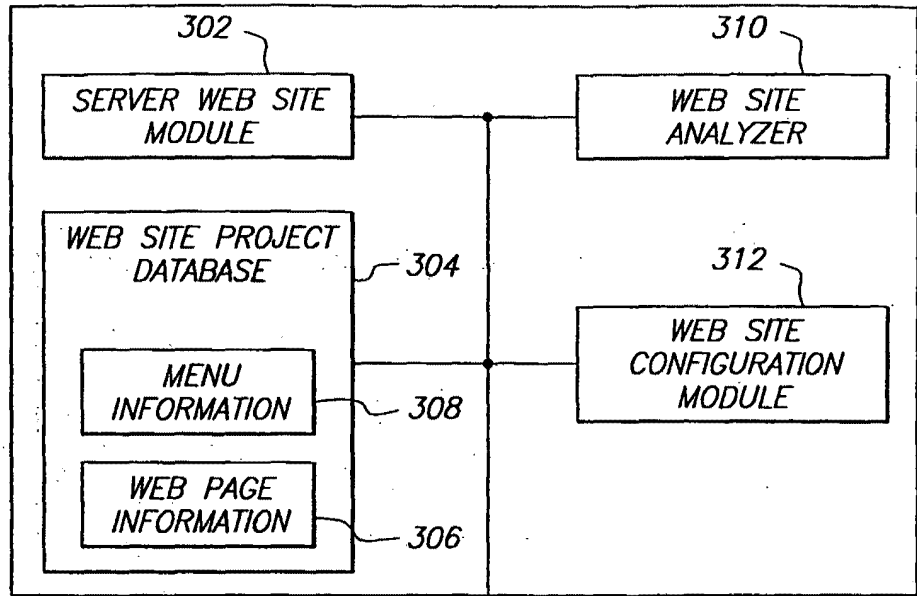


FIG. 3

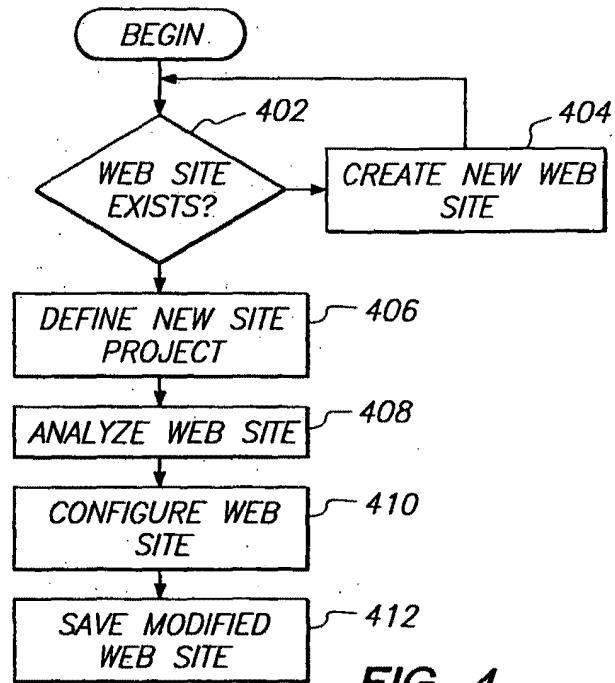


FIG. 4

SUBSTITUTE SHEET (RULE 26)

3/17

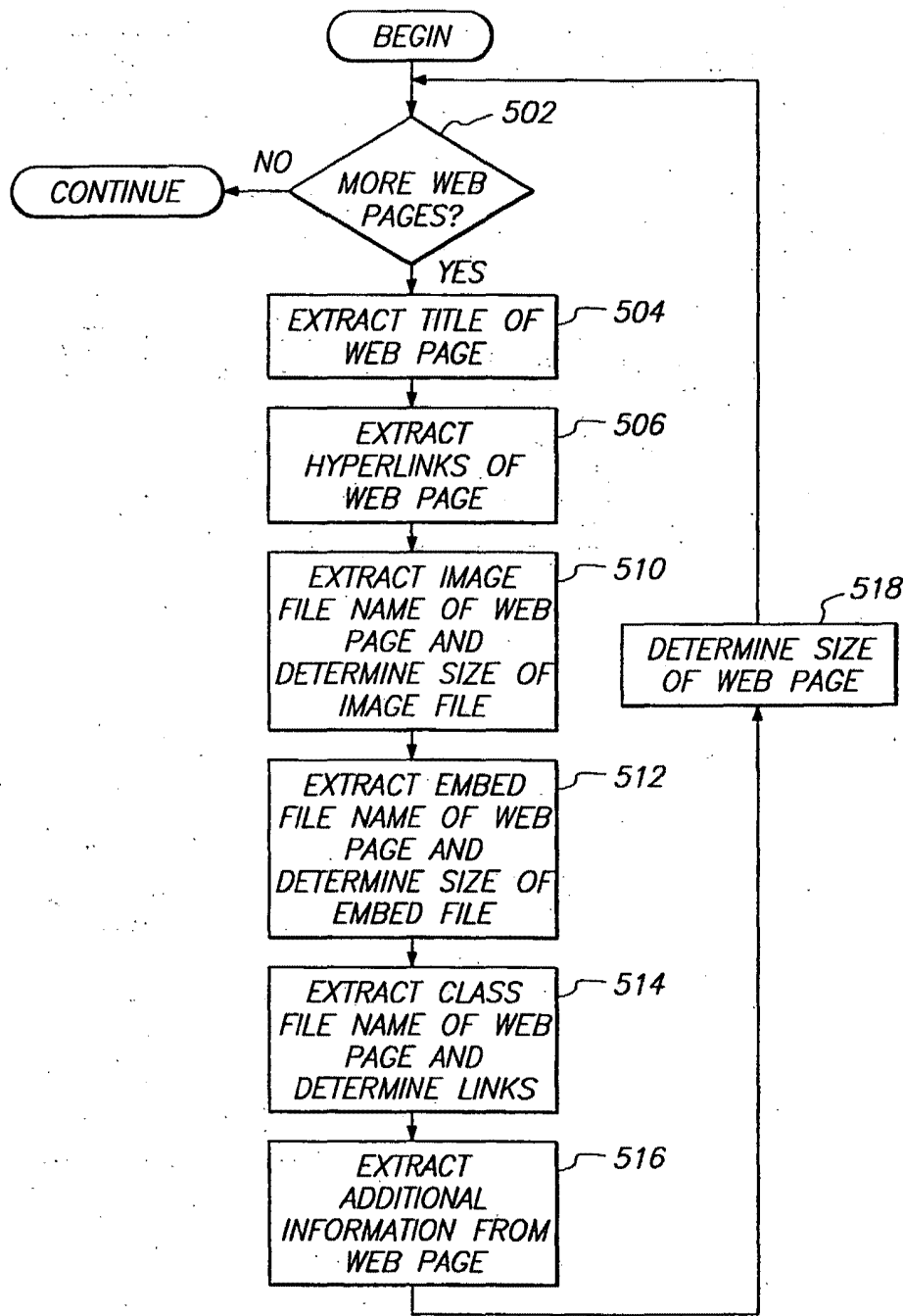
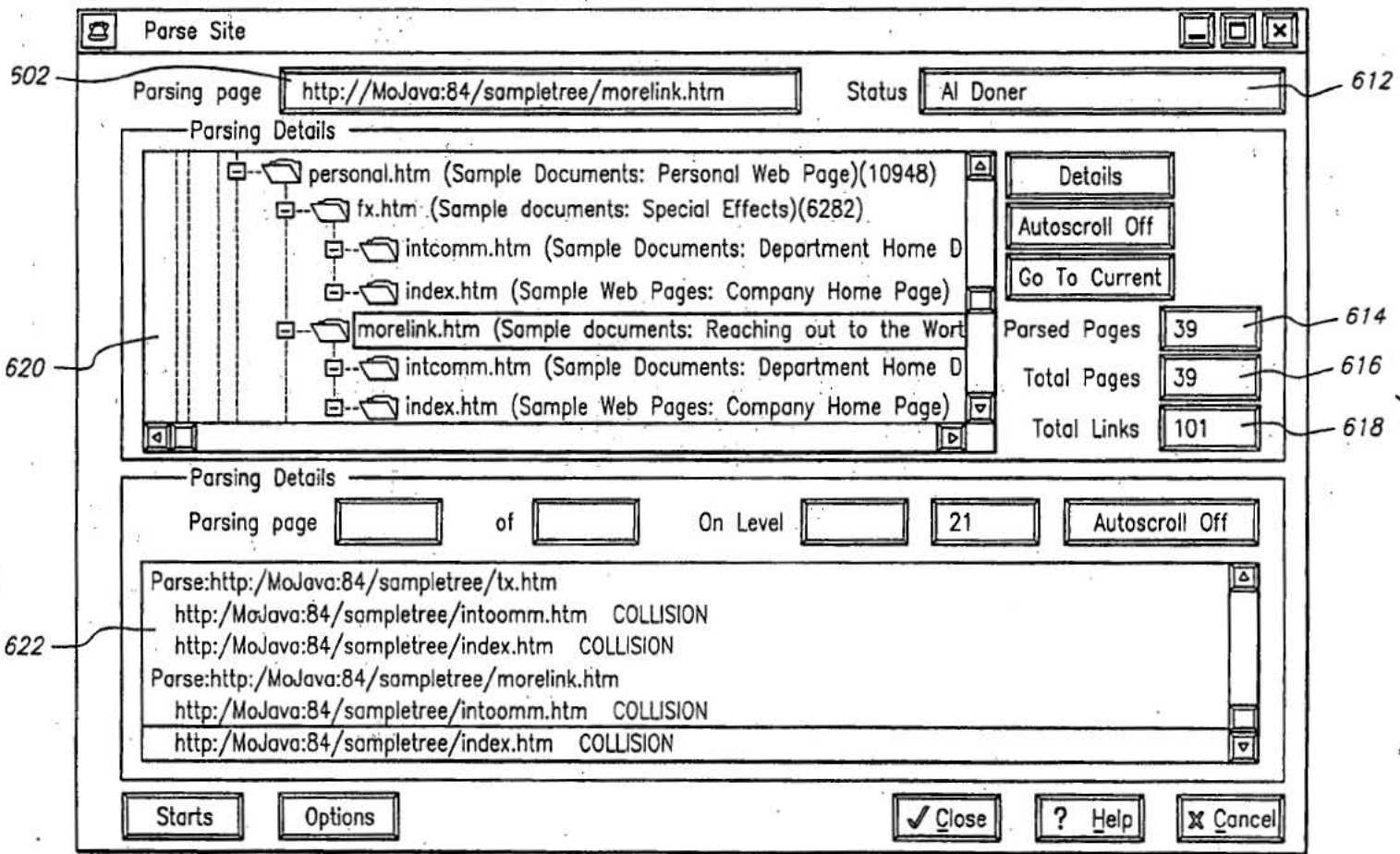


FIG. 5

SUBSTITUTE SHEET (RULE 26)



4/17

FIG. 6

5/17

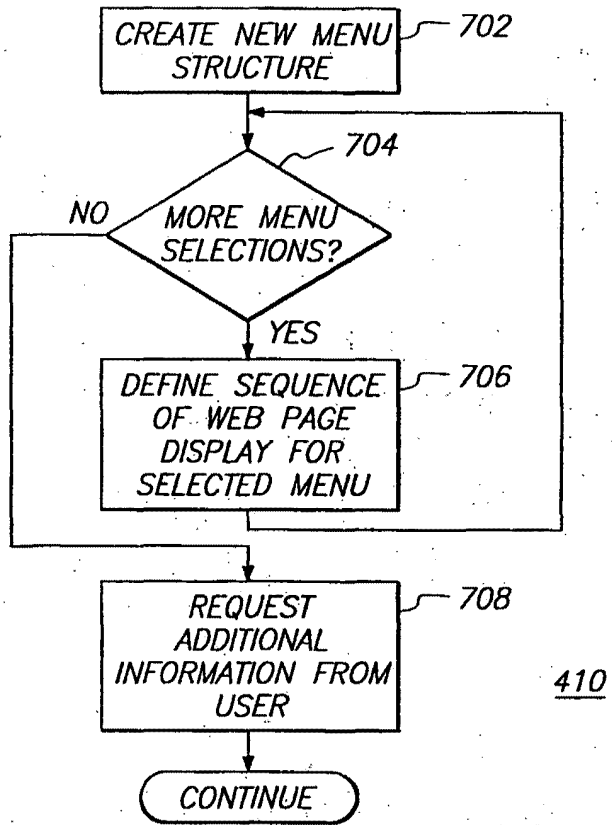


FIG. 7

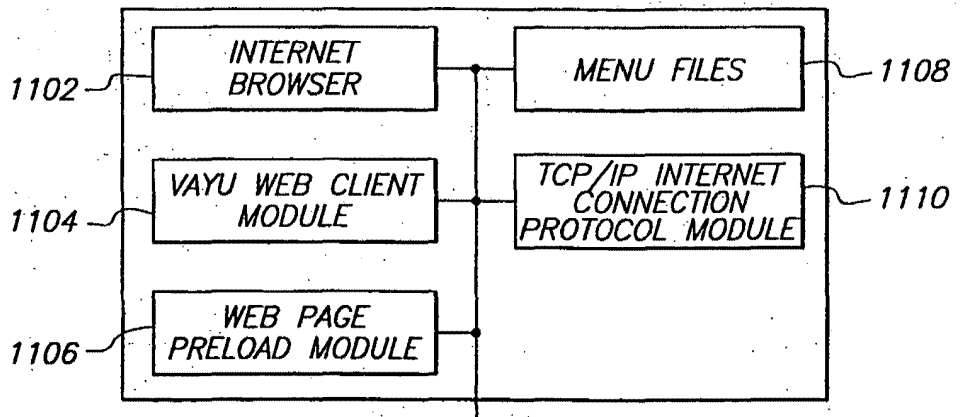


FIG. 11

SUBSTITUTE SHEET (RULE 26)

SUBSTITUTE SHEET (RULE 26)

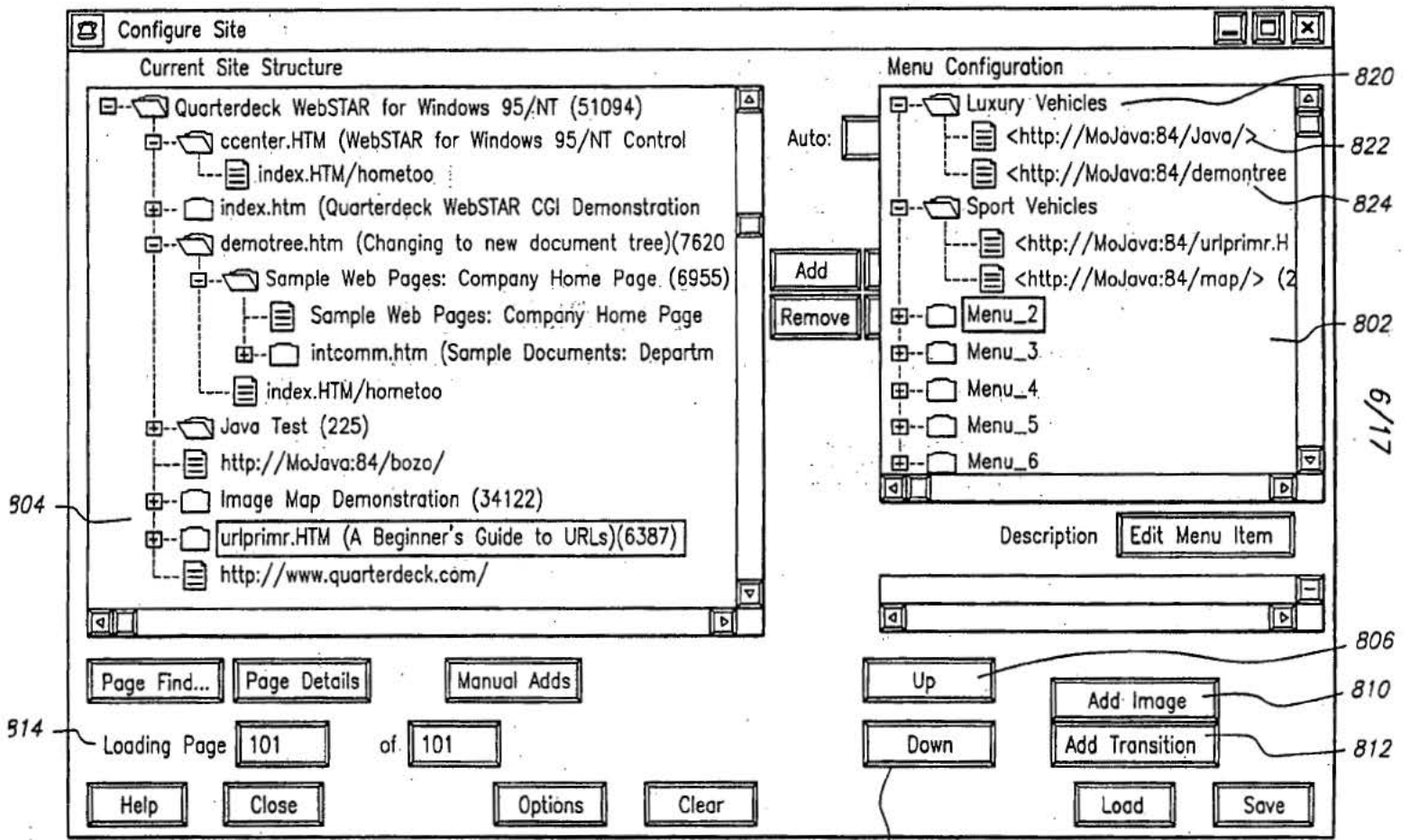


FIG. 8

7/17

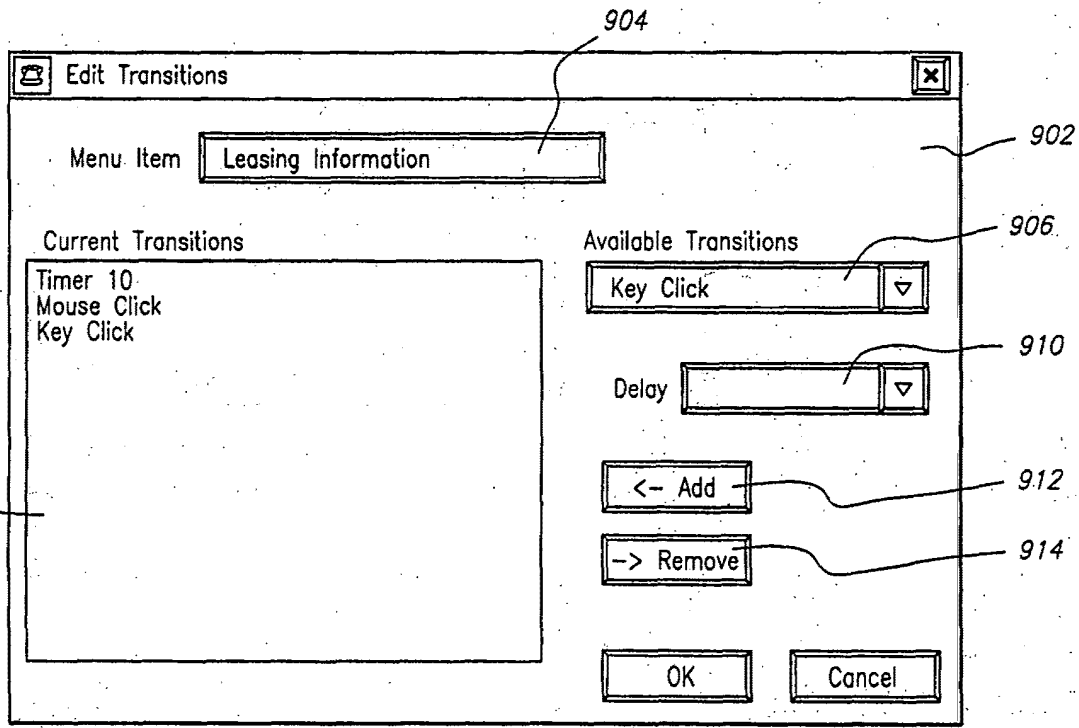


FIG. 9

SUBSTITUTE SHEET (RULE 26)

8/17

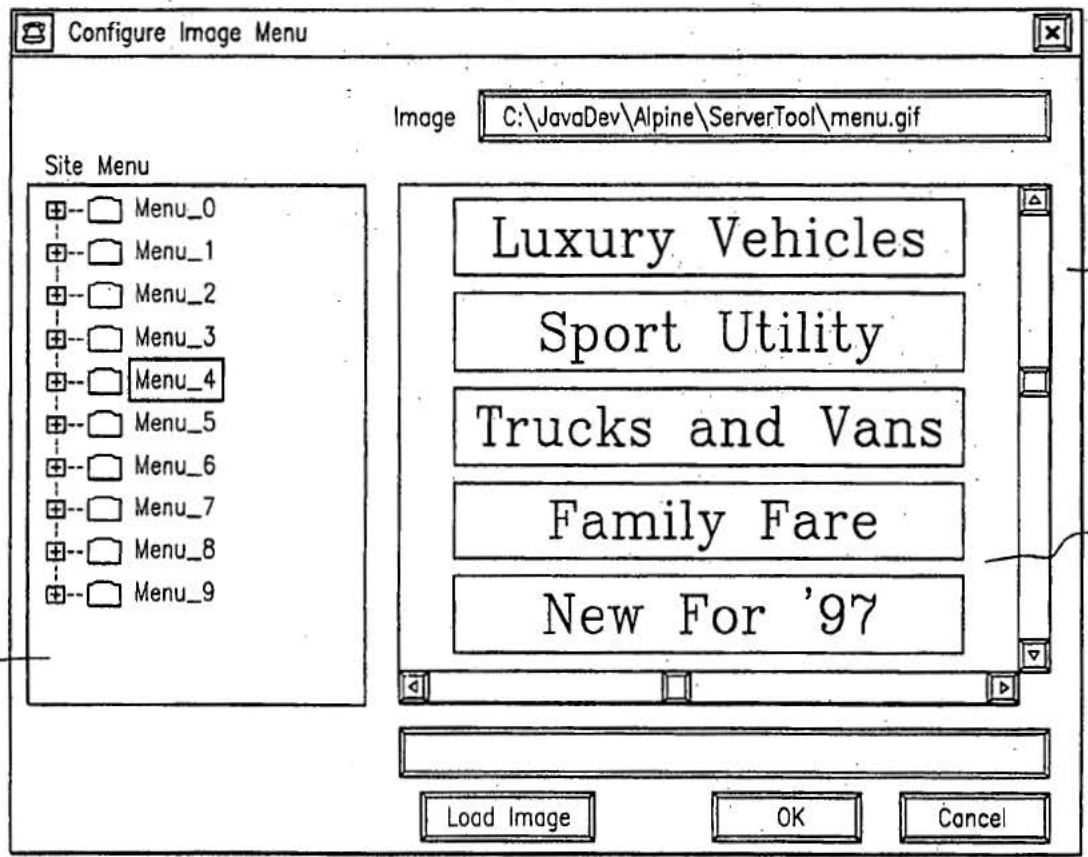


FIG. 10

SUBSTITUTE SHEET (RULE 26)

9/17

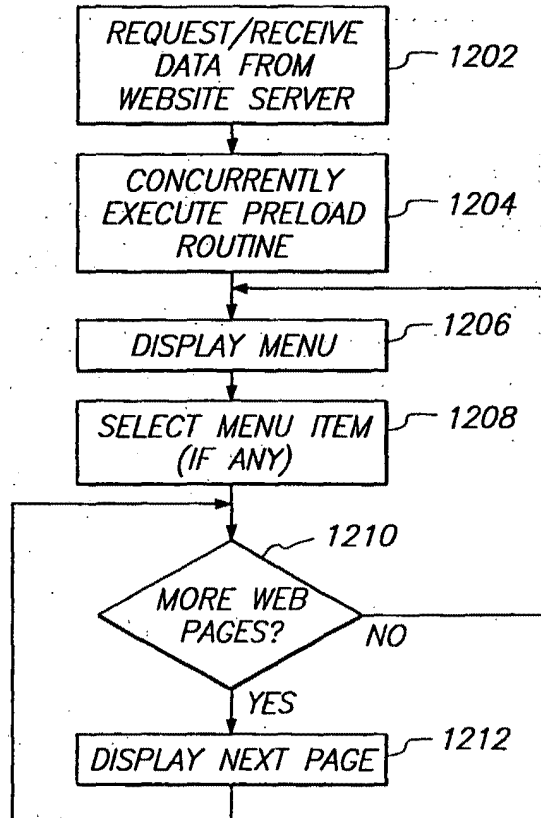


FIG. 12

SUBSTITUTE SHEET (RULE 26)

10/17

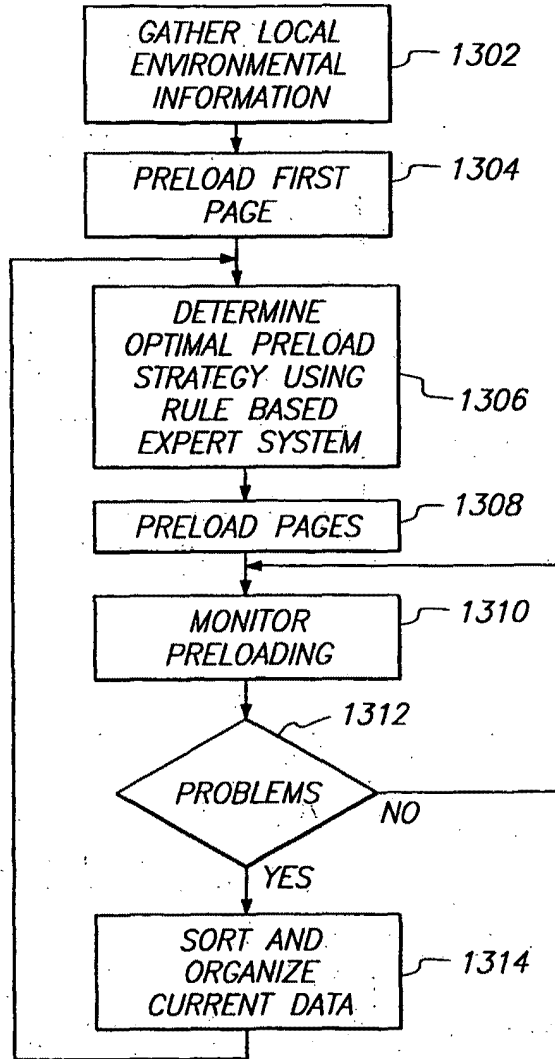


FIG. 13

SUBSTITUTE SHEET (RULE 26)

11/17

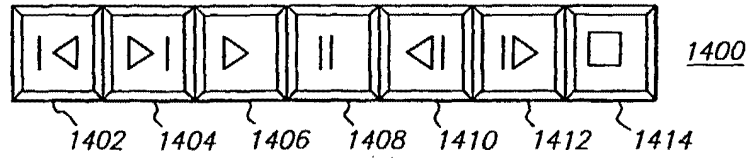


FIG. 14

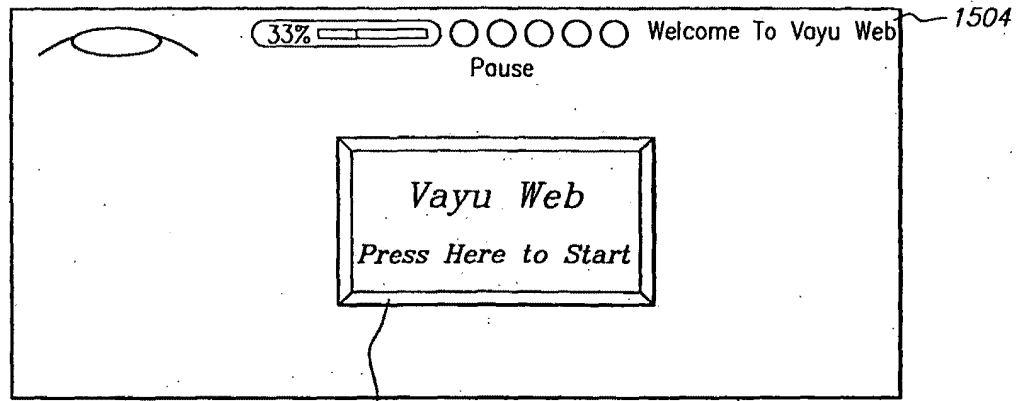


FIG. 15

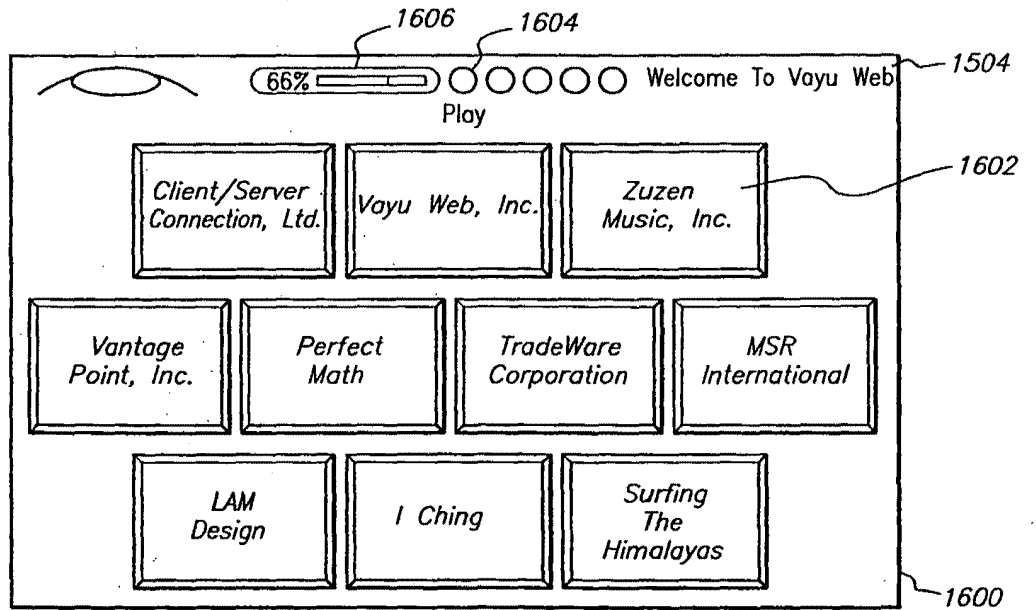


FIG. 16

SUBSTITUTE SHEET (RULE 26)

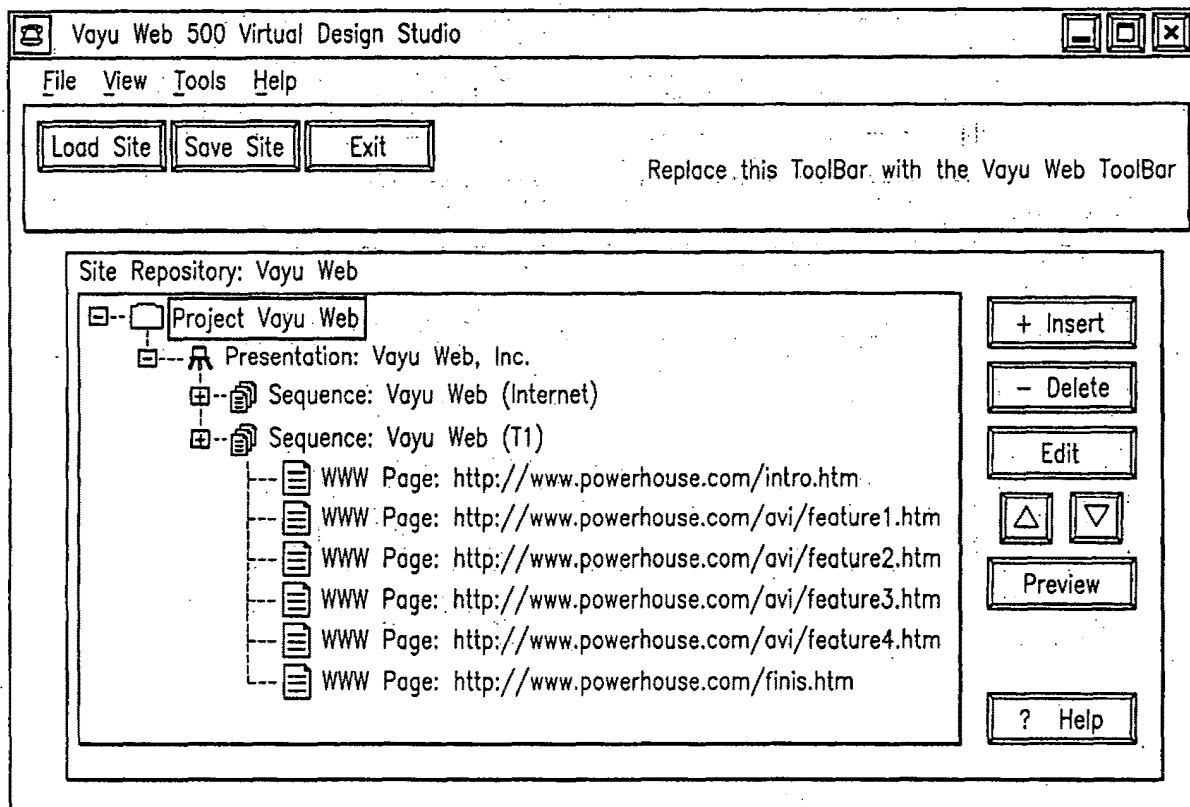


FIG. 17

SUBSTITUTE SHEET (RULE 26)

SUBSTITUTE SHEET (RULE 26)

Vayu Web Presentation Properties

Name

URL

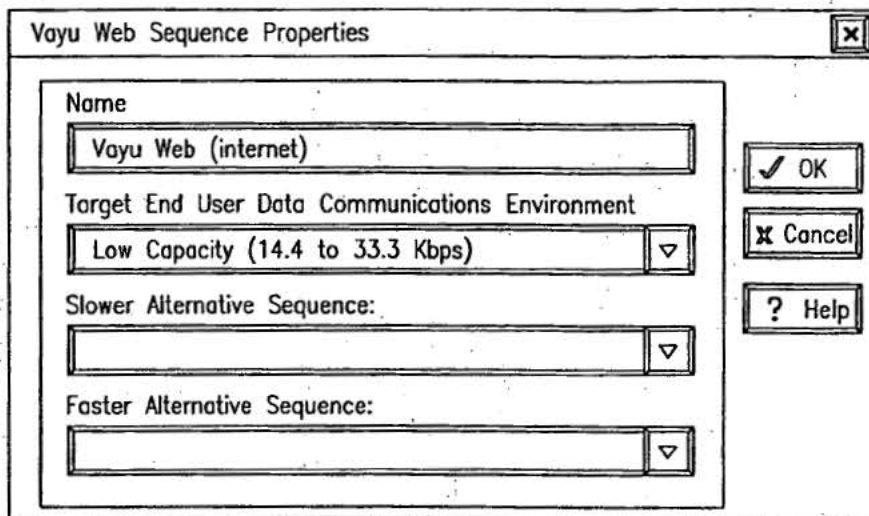
Sequences

Name	Type	Pages	Run Time
Vayu Web 500 (T1)	High Capacity	5	79 secon...
Vayu Web 500 (Mod...	Low Capacity	6	77 secon...

1802

1804

FIG. 18

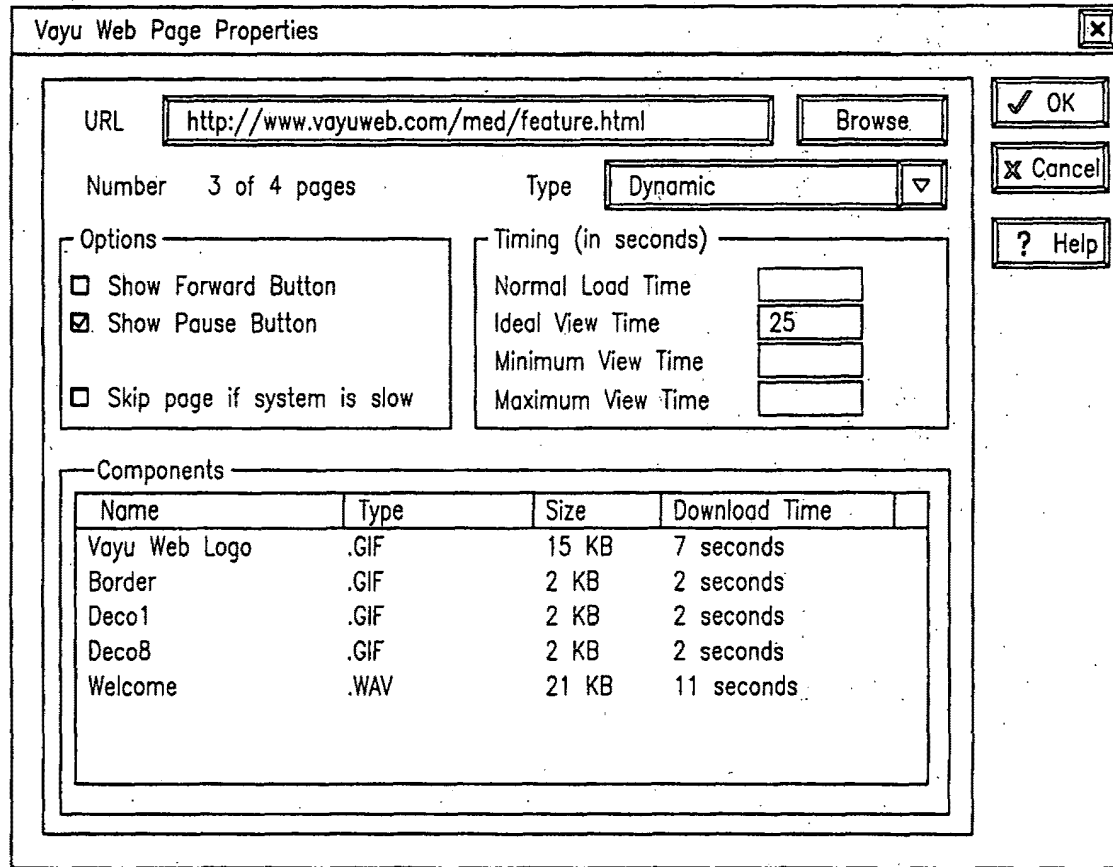


1900

FIG. 19

SUBSTITUTE SHEET (RULE 26)

15/17



2000

FIG. 20

SUBSTITUTE SHEET (RULE 26)

16/17

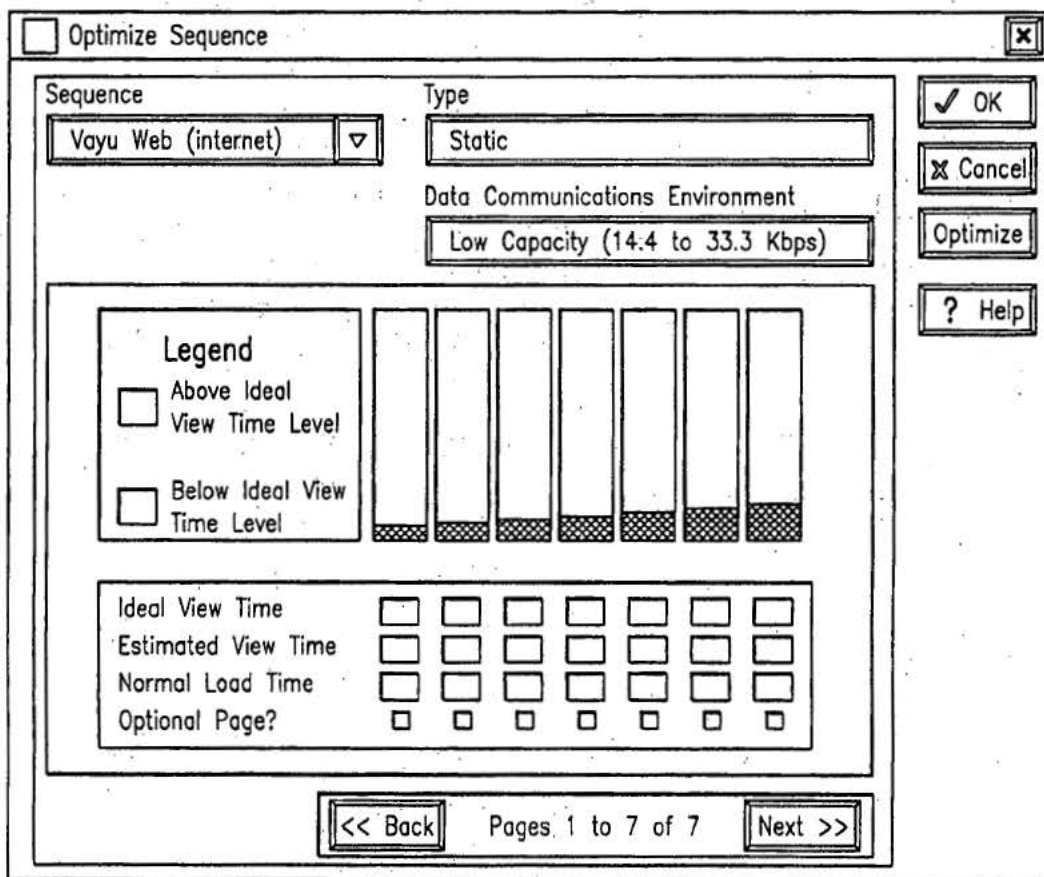


FIG. 21

SUBSTITUTE SHEET (RULE 26)

SUBSTITUTE SHEET (RULE 26)

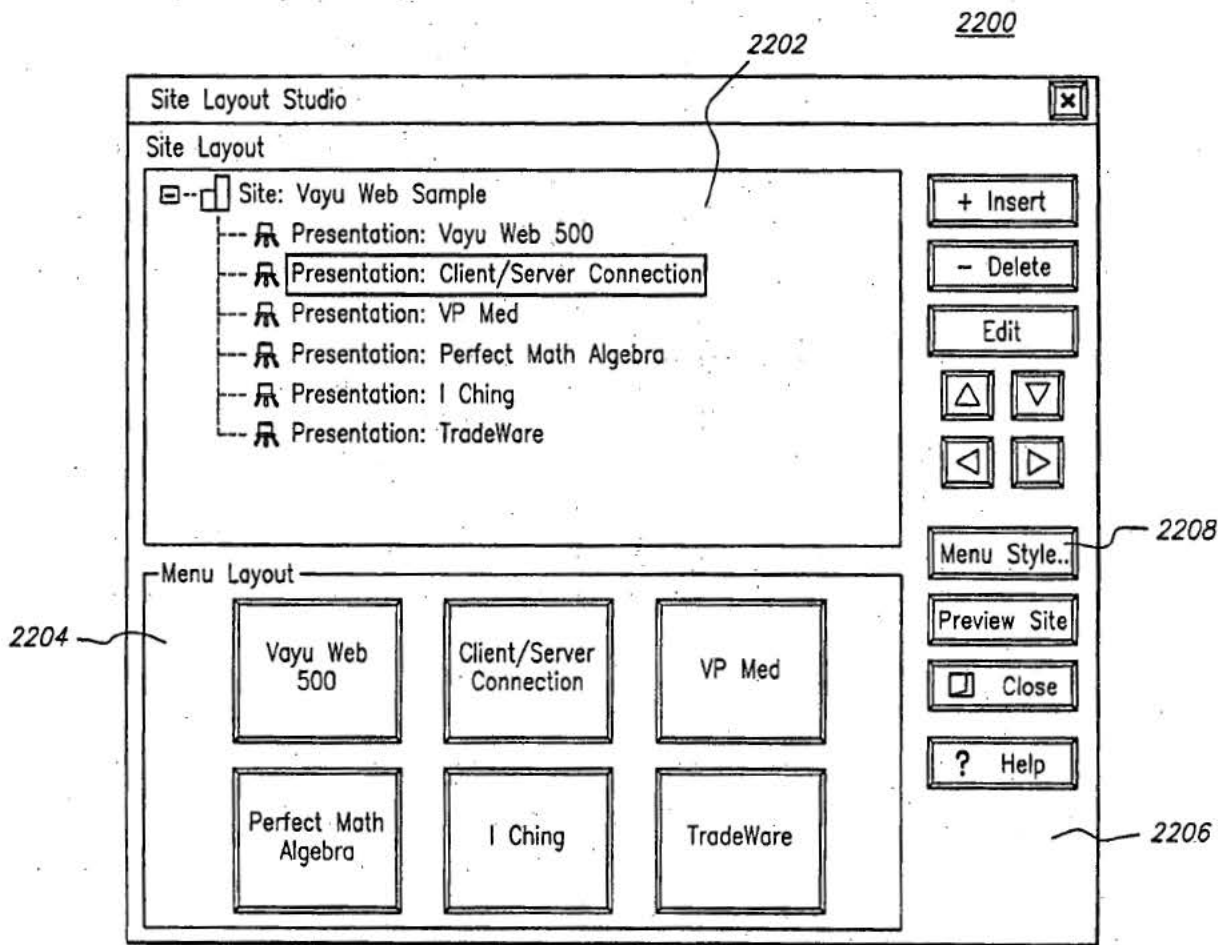


FIG. 22

17/17