

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

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
3 December 2009 (03.12.2009)

PCT

(10) International Publication Number
WO 2009/144459 A2

- (51) **International Patent Classification:**
G06Q 10/00 (2006.01)
- (21) **International Application Number:**
PCT/GB2009/001333
- (22) **International Filing Date:**
28 May 2009 (28.05.2009)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
0809933.5 30 May 2008 (30.05.2008) GB
- (71) **Applicant (for all designated States except US):** CLEAR CHANNEL UK LIMITED [GB/GB]; 33 Golden Square, London W1F 9JT (GB).
- (72) **Inventor; and**
- (75) **Inventor/Applicant (for US only):** TOMPKINS, Ian [GB/GB]; 32 Goldsworth Road, Woking, Surrey GU21 6JT (GB).
- (74) **Agents:** COZENS, Paul, Dennis et al.; Mathys & Squire LLP, 120 Holborn, London EC1N 2SQ (GB).
- (81) **Designated States (unless otherwise indicated, for every kind of national protection available):** AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) **Designated States (unless otherwise indicated, for every kind of regional protection available):** ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).
- Published:**
— with declaration under Article 17(2)(a); without abstract; title not checked by the International Searching Authority



WO 2009/144459 A2

(54) **Title:** RESOURCE-AVAILABILITY SYSTEM AND METHOD

(57) **Abstract:**

Resource-availability system and method

5 The present invention relates to apparatus for and methods of: determining the availability of a resource; processing a resource availability value map array; constructing a resource time-availability query; processing a resource availability value map in dependence on a resource time-availability query; matching a record to an array of pre-existing records according to matching criteria; processing matching period value maps; determining the extent to which the availability of a resource matches a resource availability request; and modifying a time-availability query.

10 In particular, the present invention relates to the management of resources, the availability of which is allocated typically according to finely-grained time divisions. The invention has particular, although not exclusive, relevance to the management of digital displays as used for displaying advertisements or other content which can be time-varying.

15 Public-area advertisements have traditionally involved flat-panel billboards. Often these comprise paper-based posters pasted onto a supporting surface. Although relatively cheap, such billboards are static; changing the display involves visiting the site of the billboard to physically replace the poster. Some billboards use motorised slats to display different advertisements (from a necessarily limited selection) in rotation. Other forms have a number of advertisements placed on a roll, allowing for advertisements to be scrolled into place in sequence. Both of these types of billboards nevertheless also require site visits to update the advertisements and also for regular maintenance (particularly as, being mechanical, the moving parts require frequent servicing). There is therefore great interest in replacing traditional billboards with modern digital display screens.

25 While a digital advertising panel is more expensive to install than a poster panel, it also offers numerous advantages.

30 Digital displays can be updated remotely, which is both faster and cheaper than having to deploy workers to the display location to physically replace one poster with another.

35 Digital displays can display advertisements with non-static content such as animated images and/or video. This allows them to be used to display real-time information such as time-limited special offers, dynamic betting odds, results of sporting events, lottery number draws etc. This makes the advertising panels more attractive to existing advertisers and also attracts new business from companies which would not previously have considered using static advertisements.

40 Digital panels also allow for different advertisements to be displayed according to the time of day, and for particular advertisements to be displayed only at particular times of the day. This is known in the art as 'daypart targeting'. The impact of advertisements can thus be maximised by tailoring them to the potential audience and/or the likely interests of the audience at that time (for example, different advertisements can be shown to morning versus evening commuters). This can also allow for panels to display advertisements only when the premises in which they are located are open, as, for example, with bars or public houses, which may have restricted opening hours, for example closing for a period in the afternoon.

45 The ability to display different advertisements at different times also means that different advertisers can use the same display at different times thereby maximising the usefulness of the display. Transitions between the different advertisers may be required to occur over time periods as short as a few seconds (as in a slideshow) to periods of hours or even longer. An advertising campaign may require repeat slots at predetermined (or random) time intervals over periods of days, weeks or months – as in, for example, the build-up to a product launch.

50 There is therefore a need for ways of managing the delivery of the 'digital content', in particular for managing the allocation of advertisements to particular digital panels at particular times.

55 One way to address the problem of managing the allocation of time-based resources is to provide for a generic database (or spreadsheet) detailing the resources in question (in this case, details

of the advertising panels, their characteristics, location etc); to provide, in addition to this database, information as to the availability of the resources (i.e. the panel availability); and to subsequently provide for a method of querying this database as to resource availability. However, as the number of resources (panels) increases, and as the availability of some of the resources becomes allocated, the size and complexity of the database soon increases to a point where queries can no longer be answered in real-time unless substantial computing resources have been allocated to the task.

The determination of resource availability in real-time is particularly important when, for example, negotiating the sale of advertising panels directly with a client. Real-time determination of resource availability allows for interactive negotiation which may include the discussion of scenarios and potential sales.

The present invention therefore aims to alleviate some of these problems by providing a system by which the availability of resources (such as time slots on multiple digital displays) can be determined and allocated in real-time.

Resource availability bitmap

According to the present invention there is provided apparatus for determining the availability of a resource, the apparatus comprising:

means (preferably, an input device such as a keyboard and mouse) for receiving an input regarding the availability of the resource;

means (preferably, a processor and associated memory and/or storage) for determining the availability of said resource; and

means (preferably, an output device such as a display) for providing an output regarding the availability of the resource;

wherein the means for determining the availability of the resource is adapted to determine the availability of the resource by means of a resource availability value map.

By representing the availability of the resource by means of a value map, rather than requiring the querying of a database or lookup in a table, this can permit queries of availability to be performed quickly, in real-time. This can be particularly advantageous for large datasets representing the availability of large numbers of resources (for example, hundreds or thousands of resources) over long or finely-grained timescales (for example, periods of seconds extending over several years) and/or involving complex availability queries (for example, requiring a particular duration of availability at a particular time with a particular repetition, such as availability for a minute, repeatedly hourly every morning for two weeks).

The means for receiving an input regarding the availability of the resource may comprise a query generation interface; the means for determining the availability of said resource may comprise an Availability Engine; the means for providing an output regarding the availability of the resource may comprise an Availability Preview display.

The resource availability value map may be stored remotely from the resource.

Preferably, the resource availability value map comprises a sequence of values, the position of a value within the sequence representing a time period, and the value itself representing the availability of the resource.

The value map is preferably a bitmap, and the value may indicate availability / non-availability by means of a binary representation. For example, a value of one may indicate the resource is available; zero, that the resource is unavailable. Alternatively, a ternary or higher order representation may be used, with the value preferably indicating three or more levels of availability. Hence, in general, each value in the value map may be a numeric representation of availability (including the degree of availability).

Preferably, the values, typically comprising the resource availability value map, are arranged within the sequence in chronological order.

Preferably, the first and last positions of the sequence define a time window which includes past and future time periods. By including past time periods in the resource availability value map, this can permit the historical availability and use of resources to be determined, thereby facilitating operations such as capacity planning.

5

Preferably, the apparatus further comprises means (preferably, a processor and associated memory and/or storage) for periodically evolving the position of the values within the sequence, so as preferably to maintain the approximate position of the time window about the current time period.

10

Preferably, the evolution means is adapted to maintain the approximate position of a time window, delimited by a first position and a second position in the sequence, with respect to a third position in the sequence.

15

Preferably, the third position in the sequence represents the current time period.

The periodic evolution of the sequence preferably comprises: adding new positions to the sequence at the end corresponding to future times; discarding positions at the end corresponding to past times; and repositioning the other values in the sequence to positions towards the end corresponding to past times. Hence, preferably, the evolution means further comprises means for: adding new positions to the sequence at the end corresponding to future times; discarding positions at the end corresponding to past times; and repositioning the other values in the sequence to positions towards the end corresponding to past times.

20

25

By evolving the resource availability value map in this way, the size of the resource availability value map is prevented from excessively increasing in size, thereby reducing the computing resources required.

30

Preferably, where the availability concerns that of more than one resource, the resource availability value maps are arranged to form a two-dimensional array. Preferably, the determining means is adapted to process a plurality of resource availability value maps in a two-dimensional array. Preferably, the value maps are arranged in sequence, and the apparatus further comprises means for linking the value map for a particular resource to its position within the sequence.

35

The two-dimensional array of resource availability value maps may be stored remotely from the resources.

Preferably, the apparatus further comprises means, such as a bitmap generator, for generating the resource availability value map from a resource database.

40

Preferably, the apparatus further comprises means, such as a resource selector, for generating resource selection templates (or selection filters) for selecting at least one resource. Preferably, the resource selection template is generated from resource group information or group membership information. Preferably, the resource selection template is generated from resource group information or group membership information, preferably obtained from a resource template (or filter) library. That is, resources may be selected according to information about resource properties or characteristics which can be used to define a group.

45

The resource group information or group membership information preferably comprises information regarding physical characteristics or attributes of the resource, such as the dimensions or colour depth of a digital display panel. Alternatively, the group information or group membership information may also comprise information regarding administrative characteristics or attributes of the resource, such as details of the ownership or maintenance responsibility of the resource.

50

55

Preferably, the resource selection templates or filters comprise a sequence of values, wherein preferably each value represents a resource identifier.

Preferably, the means for determining the availability of said resource includes means for identifying the resource by the position of the value in the sequence or, alternatively, by the value itself.

5 By generating resource selection templates this can permit group-specific resource availability value map arrays to be generated for a specific subset of resources, which may thereby reduce computational processing requirements and can increase processing speed.

10 Preferably, the means for generating resource selection templates includes means for combining selection templates. By combining resource selection templates in this way, more complex resource selections can be made.

15 Preferably, the apparatus further comprises a resource availability value map library, for storing group-specific resource availability value map arrays.

Preferably, the resource availability value map array is stored as a sequence of record elements, each element preferably representing the availability of the resources during a time period, each element preferably comprising at least one of:

- 20 ▪ a start period field, representing the start of the time period
- an end period field, representing the end of the time period
- a value map field, representing the availability of the resources during the time period.

25 Preferably, the record element further comprises a period lock field. Preferably, the period lock field is a mutual-exclusion or mutex lock. By using a period lock field, data errors due to the updating of the resource availability value map by more than one user or process simultaneously can be prevented. Preferably, the period lock of a record element is set by the first user or process accessing the record element, and second or subsequent users or processes attempting to access the record element are denied access until the first user or process unsets the period lock.

30 Preferably, the input receiving means is adapted to receive an input in the form of a query as to whether there are resources available to fulfil a resource requirement specified in the query. Alternatively, the query may be a generic query to determine what resources are available, without requesting specifying a resource requirement

35 Preferably, the input means includes means for accepting a resource booking request.

40 Preferably, the means for accepting the resource booking request is operable to set at least one value in the resource availability value map for the corresponding resource, for the corresponding time period specified in the booking request, preferably subject to the resource being available.

Preferably, the apparatus further comprises means for representing booked resources in a value map.

45 Preferably, the apparatus further comprises means for representing resources for which exist options (that is, preferably resources which are provisionally reserved or booked or in some other way potentially unavailable) in a value map.

50 By using additional option value maps to represent resources which are provisionally booked, this can permit differentiation between different degrees of resource availability. This can also permit responses to availability queries to comprise further information as to whether an availability request could potentially be met by available resources alone or whether it would require overriding one or more provisional bookings.

55 Preferably, each option value map represents the availability of a resource which has been booked with a predetermined degree of certainty; preferably, all options within the option value map having equal value.

60 Alternatively, options within the option value map may have a plurality of values. A plurality of option value maps may be used, each option value map representing resources which have been

booked with a degree of certainty specific to that option map. This can allow for options of different values. The value of an option may therefore reflect the booking certainty. Hence, a range of options can be envisaged, spanning the certainty space from tentative to near-certain.

- 5 The degree of certainty of a booking, or booking certainty, may depend on a characteristic of the entity making the booking. For example, certain users of a system comprising the apparatus may only be granted the ability to provisionally book resources, subject to confirmation by a supervisor. Such bookings will therefore have an initial value of booking certainty which is changed upon confirmation. Alternatively, or additionally, the booking certainty may depend on a
- 10 characteristic of the resource. For example, the resource may only come into existence after a certain time, and so a booking of that resource may only be confirmed after that time. The booking certainty may depend on the time period booked. For example, certain time periods may be designated wherein resources may only be provisionally booked. An example of this would be where no resources were permitted to be booked more than two weeks in advance. Preferably,
- 15 the degree of certainty of a booking is determined by a combination of the above listed factors.

The booking certainty is preferably represented by a value; the availability query may comprise an option override value. Preferably, the determining means is adapted to determine whether or not a provisional booking may be overridden by comparison of the option override value with the

20 booking certainty value.

Preferably, the apparatus further comprises data synchronising means for synchronising the availability data between the resource availability value maps and the resource database.

25 **Memory scrubbing**

According to a further aspect of the invention there is provided apparatus for processing a resource availability value map array, the apparatus comprising:

- 30 means (preferably, a processor and associated memory and/or storage) for comparing at least two record elements of the value map;
- means (preferably, a processor and associated memory and/or storage) for determining whether said record elements can be validly combined; and
- 35 means (preferably, a processor and associated memory and/or storage) for combining said record elements if they can be validly combined.

By processing a resource availability value map in this way, the size of the resource availability value map array or bitmap may be reduced and therefore the amount of memory required to store the bitmap can be reduced, reducing computational costs and speeding processing.

- 40 Preferably, the apparatus for processing the resource availability value map array is a memory scrubber of an Availability Engine.

In a preferred aspect of the present invention the determining means is adapted to determine that the record elements can be validly combined if at least one (preferably both) of the following

45 conditions are met:

- i) the record elements refer to adjacent time periods;
- ii) the record elements indicate identical resource availabilities.

50 Preferably, the determining means is adapted to process the resource availability value map array repeatedly.

Preferably, the determining means is adapted to process the resource availability value map according to predetermined conditions.

- 55 Preferably, the determining means is adapted to process the resource availability value map according to a schedule.

Alternatively, or in addition, the determining means is adapted to process the resource availability value map according to an internal triggering event. Such an internal triggering event may be, for

60 example, a change in the value map array detected by the apparatus. Processing the value map

array automatically can therefore maintain the value map array at an optimum size to enhance performance without requiring user intervention or an external triggering event.

5 Alternatively, the determining means is adapted to process the resource availability value map according to an external triggering event. Such an event may be in the form of a command issued by the user or by some further apparatus.

The means for combining said record elements preferably combines first and second record elements by:

10 i) setting the value of the end period field of the first record element to the value of the end period field of the second record element; and
ii) disposing of the second record element.

15 Preferably, the means for comparing record elements compares record elements two at a time; and, preferably, where the means for determining determines that the record elements may be validly combined, the means for combining record elements subsequently combines the two record elements.

20 Alternatively, the means of comparing may compare a plurality of record elements at a time; and, preferably, where the means for determining determines that the record elements may be validly combined, the means for combining record elements may subsequently combine the plurality of record elements. The plurality of record elements may be combined at the same time or, alternatively, sequentially by means of an iterative process.

25 **User interface i) Construction of the resource availability query**

According to a further aspect of the invention there is provided apparatus for constructing a resource time-availability query, the apparatus comprising:

30 means (preferably, a processor and associated memory and/or storage and/or a display) for generating a user interface comprising a time-based representation;
means (preferably, a processor and associated memory and/or storage and/or a display; an input device such as a keyboard and mouse) for determining, from the interaction of the user with the interface, the required parameters for the resource time-availability query; and
35 means (preferably, a processor and associated memory and/or storage) for constructing a resource time-availability query from said determined parameters; wherein
the required parameters comprise resource availability start and end times.

40 By using a time-based representation to construct the resource time-availability query, the user can form a query quickly, and also be spared the necessity of learning potentially complex query syntax.

Preferably, the apparatus for constructing the resource time-availability query is an availability query generator of an Availability Engine.

45 Preferably, the generating means includes means for permitting the user to adjust the availability start and end times on the time-based representation.

Preferably, the time-availability query further comprises at least some, preferably all, of the following parameters:

50

- target number of resources
- duration of required availability
- availability repetition interval
- availability date

55 Preferably, the time-based representation interface comprises one or more of:

- a date indicator
- a time-of-day indicator
- offset selector

Preferably, the determining means is operable to determine, from the user interaction with the date indicator, the dates for which availability is to be queried. A range of dates may be determined. Alternatively, individual dates may be selected.

5 Preferably, the time-based representation interface further comprises a time-of-day indicator. The time-of-day indicator is preferably divided into zones, each zone demarcating a time period of availability. Preferably, the time-of-day indicator comprises first and second user-adjustable markers, and the determining means determines the start and end times from the position of the first and second markers respectively. Alternatively, further user-adjustable markers may be
10 provided. This can permit multiple start and end times to be defined.

The time-of-day indicator may comprise a clock-face; the user-adjustable markers may comprise radii. The radii may therefore divide the clock-face into segments or zones, each segment demarcating a time period of availability.
15

Alternatively, the time-of-day indicator represents time in a non-linear representation, wherein the zones dividing the time-of-day indicator are of unequal size. By using a non-linear time representation, particularly for important time periods, this can enable the user to set one or both of the start and end times with increased precision.
20

Preferably, the time-based representation interface further comprises an offset selector for selecting an offset time value, wherein the offset time value selected by the user can allow for the start and end times to be offset by the selected time value. This may allow for the use of more finely-grained time periods than are possible by use of the time-of-day indicator alone due to resolution limitations.
25

Preferably, the generating means is operable to prevent the user from setting the availability start and end times within certain time periods. This can permit access to availability information (and resource booking facilities) to be restricted for reasons of security and/or convenience.
30

Preferably, the means for determining permits a target or total number of resources to be specified. Alternatively, the means for determining may permit specific resources to be specified by means of resource selection templates. Preferably, resource selection templates may be combined to generate more complex resource queries.
35

Preferably, the apparatus further comprises means for submitting the query for further processing. Further processing may comprise evaluating the resource availability in accordance with the query.
40

40 Pattern matching

According to a further aspect of the invention there is provided apparatus for processing a resource availability value map in dependence on a resource time-availability query, the apparatus comprising:

45 means (preferably, a processor and associated memory and/or storage) for receiving a resource availability value map;

means (preferably, a processor and associated memory and/or storage and/or a display; an input device such as a keyboard and mouse) for receiving a resource time-availability query; and

50 means (preferably, a processor and associated memory and/or storage) for matching the requirements of the resource time-availability query to the resource availability represented by the resource availability value map.

By using a resource availability value map to match the requirements of a resource time-availability query to resource availability, matches can be made quickly, in real time.
55

Preferably, the apparatus for processing the resource availability value map in dependence on a resource time-availability query is a pattern matcher of an Availability Engine.

Preferably, the matching means further comprises means for outputting the result of at least one match to at least one matching period value map.

5 Preferably, the matching period value map is a matching period map as created by a pattern matcher of an Availability Engine.

Preferably, the matching period value map comprises at least one, preferably some or all, of the following parameters:

- 10 ▪ a map start time period
- a map end time period
- a slot size
- a map status
- a map object type
- 15 ▪ the last merged start period for map
- the resource availability value map

20 Preferably, the outputting means is adapted to build a matching period value map iteratively or recursively. Preferably, the iteration is between start and end time periods. Preferably, the iteration is across all, or substantially all previously matched matching period value maps.

Preferably, for each time period, the matching means compares the resources specified in the resource time-availability query with the resource availability represented by the resource availability value map for said time period.

25 Preferably, the apparatus further comprises means for storing the matching period value maps in a matching period map store.

30 Preferably, the apparatus further comprises means for selecting the optimum matching period value map. The optimum matching period value map is the one that best matches the resources specified in the resource time-availability query. The best match may be determined according to the availability of specific resources or, alternatively, according to the availability of a total number of resources.

35 In an alternative, for faster processing, the apparatus is adapted to match a matching period value map to a subset of pre-existing matching period value maps, selected according to partial matching criteria. This can speed processing by reducing the size of the data set under consideration; rather than requiring a recursive iteration over all the matching period maps already discovered, only the subset of potentially suitable matching period maps needs to be considered.

40 This feature is provided independently. Hence, in a further aspect of the invention there is provided apparatus to match a record to members of an array of pre-existing records according to matching criteria, comprising:

- 45 means (preferably, a processor and associated memory and/or storage) for selecting a subset of the pre-existing records according to a first part of the matching criteria; and
- means (preferably, a processor and associated memory and/or storage) for comparing the record to the selected subset of the pre-existing records according to a second part of the matching criteria.

50 The record may be a matching period value map; the pre-existing records may be previously matched matching period value maps.

55 Preferably, the apparatus further comprises means for generating a parameter associated with each pre-existing record according to the first part of the matching criteria. The apparatus may further comprise means for generating a table from the pre-existing records, indexed according to the generated parameter. This can allow for the record to be rapidly matched to pre-existing records by means of a simple look-up table rather than a full iteration across all, or substantially all pre-existing records.

60 Preferably, the matching criteria are temporal criteria. The first and/or second parts of the criteria may comprise any of the following: year, season, month, date, day, hour, minute, second.

Preferably, the table may comprise pointers (such as memory pointers) to the pre-existing records.

5 **Filtering**

According to a further aspect of the invention there is provided apparatus for processing matching period value maps, the apparatus comprising:

10 means (preferably, a processor, memory and storage) for receiving a first period value map and a second period value map;

means (preferably, a processor and associated memory and/or storage) for comparing a first period value map and second period value map; and

15 means (preferably, a processor and associated memory and/or storage) for reallocating resources between the first period value map and the second period value map in dependence on at least one pre-determined requirement.

By reallocating resources between period value maps, the form of the availability data can be optimised according to requirements.

20 Preferably, the apparatus for processing the matching period value maps is a filter part of an Availability Engine.

Preferably, the requirement comprises one or more of the following:

- 25 ▪ that resources are available for the longest possible consecutive period of time
- that periods of resource availability are preferably contiguous in time and not fragmented
- that the number of available resources during a time period is constant

30 Preferably, the list of requirements may comprise one or more further sets of requirements which may be used singly or in combination, either with other members of a set or with the original set of requirements.

User interface ii) Output of resource availability results

35 According to a further aspect of the invention there is provided apparatus for determining the extent to which the availability of a resource matches a resource availability request, the apparatus comprising:

means (preferably, a processor and associated memory and/or storage) for comparing the resource availability request with the availability of the resource; and

40 and/or a display) for outputting a signal representing the extent to which the availability of the resource matches the resource availability request.

45 By outputting information in this way the user is kept informed of the extent to which the resource availability request can be fulfilled by the available resources.

Preferably, the representation signal is time-based. More preferably, the representation signal represents the resource availability as a function of time.

50 Preferably, the outputting means further comprises display means for generating a time-based representation in dependence on the representation signal.

55 Preferably, the outputting means is operable to generate a plurality of representation signals regarding the degree to which the resource availability request can be fulfilled by the available resources at a plurality of times.

Preferably, the display means is operable to generate a time-based representation in dependence on the plurality of representation signals and each representation signal is represented in an area of the time-based representation.

Preferably, the area is further divided into zones according to the variation of resource availability during the time period specified in the resource availability request. Each zone may indicate different resource availability.

- 5 Preferably, the outputting means allocates to each zone of the area an attribute according to whether the resource availability request can be:
- met, with 50%, 20%, 10%, 5%, or 1% spare resource capacity; or
 - met, with less than 20%, 10%, 5%, or 1% spare resource capacity; or
 - not met, but within 10%, 5%, or 1% of the requested resources; or
- 10 ▪ not met, and by more than 1%, 5%, 10%, 20%, or 50% of the requested resources

15 Preferably, the outputting means is further operable to generate data regarding the degree to which the resource availability request could be fulfilled by resources on which there exists an option. Preferably, the outputting means may output a signal representing this data. The display means may generate a time-based representation in dependence on this signal in a further area of the time-based representation.

20 Preferably, the further area is further divided into zones according to the variation of resource options during the time period specified in the resource availability request.

25 Preferably, the outputting means allocates to each zone of the further area an attribute according to whether or not the resource availability request can be met by overriding options. The attribute may take binary values, wherein one of the binary values indicates options which may be overridden, and the other binary value indicates options which may not be overridden. In an alternative, a range of attributes may be used, indicating a hierarchy of options.

30 Preferably, the outputting means is further operable to generate data regarding the degree to which the resource availability request could be fulfilled if the total number of resources were available and to output a signal representing this data. The display means may generate a time-based representation in dependence on this signal in a yet further area of the time-based representation.

35 In an alternative, the outputting means generates only one area of the time-based representation. Alternatively, only two areas of the time-based representation are generated.

40 Preferably, the areas are circular. Preferably, the zones are segments of a circle. Preferably, the circles are concentric. Preferably, the time-based representation is in the form of a clock. Preferably, the clock is marked into twelve-hour divisions, more preferably twenty-four-hour divisions.

Alternatively, the time-based representation may have a non-circular form, for example a line or other shape. Alternatively, the time-based representation represents time in a non-linear representation.

45 Preferably, the apparatus further comprises means for processing a plurality of period value maps and selecting the period value map which best matches the resource availability request.

50 Preferably, the outputting means is operable to generate data relating to the period value maps processed and to output a signal representing this data. The display means may generate a table in dependence on this signal

User interface *iii) Drag & drop feature*

55 According to a further aspect of the invention there is provided apparatus for modifying a time-availability query, the apparatus comprising:

- means (preferably, a processor and associated memory and/or storage, and/or an input device such as a keyboard and mouse) for receiving a first time-availability query;
 - means for receiving a resource selection template; and
 - means for generating a second time-availability query by applying the resource selection template to the first time-availability query.
- 60

By providing a way for a time-availability query to be modified by application of a resource selection template this can enhance interactivity.

- 5 Preferably, the apparatus further comprises means for further processing the query. Further processing may comprise evaluating the resource availability in accordance with the query.

Preferably, the query may be further modified by applying further resource selection templates, each successive modification being evaluated in turn.

10

Corresponding methods

According to a further aspect of the present invention there is provided a method of determining the availability of a resource, the method comprising:

15

- receiving an input regarding the availability of the resource;
- determining the availability of said resource; and
- providing an output regarding the availability of the resource;

wherein the method of determining the availability of the resource comprises determining the availability of the resource by means of a resource availability value map.

20

Preferably, the resource availability value map comprises a sequence of values, the position of a value within the sequence representing a time period, and the value itself representing the availability of the resource.

25

The value map is preferably a bitmap, and the value may indicate availability / non-availability by means of a binary representation. For example, a value of one may indicate the resource is available; zero, that the resource is unavailable. Alternatively, a ternary or higher order representation may be used, with the value preferably indicating three or more levels of availability. Hence, in general, each value in the value map may be a numeric representation of availability (including the degree of availability).

30

Preferably, the values, typically comprising the resource availability value map, are arranged within the sequence in chronological order.

35

Preferably, the first and last positions of the sequence define a time window which includes past and future time periods.

Preferably, the method further comprises a method of periodically evolving the position of the values within the sequence, so as preferably to maintain the approximate position of the time window about the current time period.

40

Preferably, the method of evolution maintains the approximate position of a time window, delimited by a first position and a second position in the sequence, with respect to a third position in the sequence.

45

Preferably, the third position in the sequence represents the current time period.

The periodic evolution of the sequence preferably comprises: adding new positions to the sequence at the end corresponding to future times; discarding positions at the end corresponding to past times; and repositioning the other values in the sequence to positions towards the end corresponding to past times.

50

Preferably, where the availability concerns that of more than one resource, the resource availability value maps are arranged to form a two-dimensional array. Preferably, the method of determining processes a plurality of resource availability value maps in a two-dimensional array. Preferably, the value maps are arranged in sequence, and the method further comprises a method of linking the value map for a particular resource to its position within the sequence.

55

The two-dimensional array of resource availability value maps may be stored remotely from the resources.

60

Preferably, the method further comprises a method of generating the resource availability value map from a resource database.

- 5 Preferably, the method further comprises a method of generating resource selection templates (or selection filters) for selecting at least one resource. Preferably, the resource selection template is generated from resource group information or group membership information. Preferably, the resource selection template is generated from resource group information or group membership information, preferably obtained from a resource template (or filter) library. That is, resources may be selected according to information about resource properties or characteristics which can be used to define a group.

10 The resource group information or group membership information preferably comprises information regarding physical characteristics or attributes of the resource, such as the dimensions or colour depth of a digital display panel. Alternatively, the group information or group membership information may also comprise information regarding administrative characteristics or attributes of the resource, such as details of the ownership or maintenance responsibility of the resource.

- 15 Preferably, the resource selection templates or filters comprise a sequence of values, wherein preferably each value represents a resource identifier.

20 Preferably, determining the availability of said resource includes identifying the resource by the position of the value in the sequence or, alternatively, by the value itself.

- 25 Preferably, generating resource selection templates includes combining selection templates. By combining resource selection templates in this way, more complex resource selections can be made.

- 30 Preferably, the method further comprises storing group-specific resource availability value map arrays in a resource availability value map library.

35 Preferably, the method further comprises storing the resource availability value map array as a sequence of record elements, each element preferably representing the availability of the resources during a time period, each element preferably comprising at least one of:

- a start period field, representing the start of the time period
- an end period field, representing the end of the time period
- a value map field, representing the availability of the resources during the time period.

- 40 Preferably, the record element further comprises a period lock field. Preferably, the period lock field is a mutual-exclusion or mutex lock. Preferably, the period lock of a record element is set by the first user or process accessing the record element, and second or subsequent users or processes attempting to access the record element are denied access until the first user or process unsets the period lock.

45 Preferably, the method of receiving receives an input in the form of a query as to whether there are resources available to fulfil a resource requirement specified in the query. Alternatively, the query may be a generic query to determine what resources are available, without requesting specifying a resource requirement

- 50 Preferably, the input method includes a method of accepting a resource booking request.

55 Preferably, the method of accepting the resource booking request sets at least one value in the resource availability value map for the corresponding resource, for the corresponding time period specified in the booking request, preferably subject to the resource being available.

Preferably, the method further comprises a method of representing booked resources in a value map.

Preferably, the method further comprises a method of representing resources for which exist options (that is, preferably resources which are provisionally reserved or booked or in some other way potentially unavailable) in a value map.

5 Preferably, each option value map represents the availability of a resource which has been booked with a predetermined degree of certainty; preferably, all options within the option value map having equal value.

10 Alternatively, options within the option value map may have a plurality of values. A plurality of option value maps may be used, each option value map representing resources which have been booked with a degree of certainty specific to that option map. This can allow for options of different values. The value of an option may therefore reflect the booking certainty. Hence, a range of options can be envisaged, spanning the certainty space from tentative to near-certain.

15 The degree of certainty of a booking, or booking certainty, may depend on a characteristic of the entity making the booking. For example, certain users of the method may only be granted the ability to provisionally book resources, subject to confirmation by a supervisor. Such bookings will therefore have an initial value of booking certainty which is changed upon confirmation. Alternatively, or additionally, the booking certainty may depend on a characteristic of the resource. For example, the resource may only come into existence after a certain time, and so a booking of that resource may only be confirmed after that time. The booking certainty may depend on the time period booked. For example, certain time periods may be designated wherein resources may only be provisionally booked. An example of this would be where no resources were permitted to be booked more than two weeks in advance. Preferably, the degree of certainty
20 of a booking is determined by a combination of the above listed factors.

The booking certainty is preferably represented by a value; the availability query may comprise an option override value. Preferably, the method of determining determines whether or not a provisional booking may be overridden by comparison of the option override value with the booking certainty value.
30

Preferably, the method further comprises a method of synchronising the availability data between the resource availability value maps and the resource database.

35 According to a further aspect of the invention there is provided a method of processing a resource availability value map array, the method comprising:
comparing at least two record elements of the value map;
determining whether said record elements can be validly combined; and
combining said record elements if they can be validly combined.
40

In a preferred aspect of the present invention the method of determining determines that the record elements can be validly combined if at least one (preferably both) of the following conditions are met:

- 45
- i) the record elements refer to adjacent time periods;
 - ii) the record elements indicate identical resource availabilities.

Preferably, the determining method processes the resource availability value map array repeatedly.

50 Preferably, the determining method processes the resource availability value map according to predetermined conditions.

Preferably, the determining method processes the resource availability value map according to a schedule.

55 Alternatively, or in addition, the determining method processes the resource availability value map according to an internal triggering event. Such an internal triggering event may be, for example, a change in the value map array detected by the method.

Alternatively, the determining method processes the resource availability value map according to an external triggering event. Such an event may be in the form of a command issued by the user or by some further method.

- 5 The method of combining said record elements preferably combines first and second record elements by:
- i) setting the value of the end period field of the first record element to the value of the end period field of the second record element; and
 - 10 ii) disposing of the second record element.

10 Preferably, the method of comparing record elements compares record elements two at a time; and, preferably, where the method of determining determines that the record elements may be validly combined, the method of combining record elements subsequently combines the two record elements.

- 15 Alternatively, the method of comparing may compare a plurality of record elements at a time; and, preferably, where the method of determining determines that the record elements may be validly combined, the method of combining record elements may subsequently combine the plurality of record elements. The plurality of record elements may be combined at the same time or, 20 alternatively, sequentially by means of an iterative process.

According to a further aspect of the invention there is provided a method of constructing a resource time-availability query, the method comprising:

- 25 generating a user interface comprising a time-based representation;
determining, from the interaction of the user with the interface, the required parameters for the resource time-availability query; and
constructing a resource time-availability query from said determined parameters; wherein the required parameters comprise resource availability start and end times.

- 30 Preferably, the method of generating includes a method of permitting the user to adjust the availability start and end times on the time-based representation.

Preferably, the time-availability query further comprises at least some, preferably all, of the following parameters:

- 35
- target number of resources
 - duration of required availability
 - availability repetition interval
 - availability date

- 40 Preferably, the time-based representation interface comprises one or more of:
- a date indicator
 - a time-of-day indicator
 - offset selector

- 45 Preferably, the method of determining determines, from the user interaction with the date indicator, the dates for which availability is to be queried. A range of dates may be determined. Alternatively, individual dates may be selected.

- 50 Preferably, the time-based representation interface further comprises a time-of-day indicator. The time-of-day indicator is preferably divided into zones, each zone demarcating a time period of availability. Preferably, the time-of-day indicator comprises first and second user-adjustable markers, and the method of determining determines the start and end times from the position of the first and second markers respectively. Alternatively, further user-adjustable markers may be provided. This can permit multiple start and end times to be defined.

- 55 The time-of-day indicator may comprise a clock-face; the user-adjustable markers may comprise radii. The radii may therefore divide the clock-face into segments or zones, each segment demarcating a time period of availability.

Alternatively, the time-of-day indicator represents time in a non-linear representation, wherein the zones dividing the time-of-day indicator are of unequal size.

5 Preferably, the time-based representation interface further comprises an offset selector for selecting an offset time value, wherein the offset time value selected by the user can allow for the start and end times to be offset by the selected time value.

10 Preferably, the method of generating prevents the user from setting the availability start and end times within certain time periods.

15 Preferably, the method of determining permits a target or total number of resources to be specified. Alternatively, the method of determining may permit specific resources to be specified by means of resource selection templates. Preferably, resource selection templates may be combined to generate more complex resource queries.

20 Preferably, the method further comprises a method of submitting the query for further processing. Further processing may comprise evaluating the resource availability in accordance with the query.

25 According to a further aspect of the invention there is provided a method of processing a resource availability value map in dependence on a resource time-availability query, the method comprising:

receiving a resource availability value map;

receiving a resource time-availability query; and

25 matching the requirements of the resource time-availability query to the resource availability represented by the resource availability value map.

30 Preferably, the method of matching further comprises a method of outputting the result of at least one match to at least one matching period value map.

35 Preferably, the method of matching period value maps comprises at least one, preferably some or all, of the following parameters:

▪ a map start time period

▪ a map end time period

40 ▪ a slot size

▪ a map status

▪ a map object type

▪ the last merged start period

▪ the resource availability value map

45 Preferably, the method of outputting builds the matching period value map iteratively or recursively. Preferably, the iteration is between start and end time periods. Preferably, the iteration is across all, or substantially all previously matched matching period value maps.

50 In an alternative, the method of outputting builds a matching period value map from a subset of previously matched matching period value maps.

55 Preferably, for each time period, the method of matching compares the resources specified in the resource time-availability query with the resource availability represented by the resource availability value map for said time period.

60 Preferably, the method further comprises a method of storing the matching period value maps in a matching period map store.

65 Preferably, the method further comprises a method of selecting the optimum matching period value map. The optimum matching period value map is the one that best matches the resources specified in the resource time-availability query. The best match may be determined according to the availability of specific resources or, alternatively, according to the availability of a total number of resources.

In a further aspect of the invention there is provided a method of matching a record to an array of pre-existing records according to matching criteria, comprising:

selecting a subset of the pre-existing records according to a first part of the matching criteria; and

5 comparing the record to the selected subset of the pre-existing records according to a second part of the matching criteria.

The record may be a matching period value map; the pre-existing records may be previously matched matching period value maps.

10 Preferably, the method further comprises generating a parameter associated with each pre-existing record according to the first part of the matching criteria. The method may further comprise generating a table from the pre-existing records, indexed according to the generated parameter.

15 According to a further aspect of the invention there is provided a method of processing matching period value maps, the method comprising:

receiving a first period value map and a second period value map;

comparing a first period value map and second period value map; and

20 reallocating resources between the first period value map and the second period value map in dependence on at least one pre-determined requirement.

Preferably, the requirement comprises one or more of the following:

- that resources are available for the longest possible consecutive period of time
- that periods of resource availability are preferably contiguous in time and not fragmented
- 25 ▪ that the number of available resources during a time period is constant

30 Preferably, the list of requirements may comprise one or more further sets of requirements which may be used singly or in combination, either with other members of a set or with the original set of requirements.

According to a further aspect of the invention there is provided a method of determining the extent to which the availability of a resource matches a resource availability request, the method comprising:

35 comparing the resource availability request with the availability of the resource; and

outputting a signal representing the extent to which the availability of the resource matches the resource availability request.

40 Preferably, the representation signal is time-based. More preferably, the representation signal represents the resource availability as a function of time.

45 Preferably, the method of outputting further comprises a method of display for generating a time-based representation in dependence on the representation signal.

Preferably, the method of outputting generates a plurality of representation signals regarding the degree to which the resource availability request can be fulfilled by the available resources at a plurality of times.

50 Preferably, the method of display generates a time-based representation in dependence on the plurality of representation signals and each representation signal is represented in an area of the time-based representation.

55 Preferably, the area is further divided into zones according to the variation of resource availability during the time period specified in the resource availability request. Each zone may indicate different resource availability.

60 Preferably, the method of outputting allocates to each zone of the area an attribute according to whether the resource availability request can be:

- met, with 50%, 20%, 10%, 5%, or 1% spare resource capacity; or
- met, with less than 20%, 10%, 5%, or 1% spare resource capacity; or
- not met, but within 10%, 5%, or 1% of the requested resources; or

- not met, and by more than 1%, 5%, 10%, 20%, or 50% of the requested resources

5 Preferably, the method of outputting generates data regarding the degree to which the resource availability request could be fulfilled by resources on which there exists an option. Preferably, the method of outputting outputs a signal representing this data. The method of display may generate a time-based representation in dependence on this signal in a further area of the time-based representation.

10 Preferably, the further area is further divided into zones according to the variation of resource options during the time period specified in the resource availability request.

15 Preferably, the method of outputting allocates to each zone of the further area an attribute according to whether or not the resource availability request can be met by overriding options. The attribute may take binary values, wherein one of the binary values indicates options which may be overridden, and the other binary value indicates options which may not be overridden. In an alternative, a range of attributes may be used, indicating a hierarchy of options.

20 Preferably, the method of outputting generates data regarding the degree to which the resource availability request could be fulfilled if the total number of resources were available and to output a signal representing this data. The method of display may generate a time-based representation in dependence on this signal in a yet further area of the time-based representation.

25 In an alternative, the method of outputting generates only one area of the time-based representation. Alternatively, only two areas of the time-based representation are generated.

30 Preferably, the areas are circular. Preferably, the zones are segments of a circle. Preferably, the circles are concentric. Preferably, the time-based representation is in the form of a clock. Preferably, the clock is marked into twelve –hour divisions, more preferably twenty-four hour divisions.

Alternatively, the time-based representation may have a non-circular form, for example a line or other shape. Alternatively, the time-based representation represents time in a non-linear representation.

35 Preferably, the method further comprises a method of processing a plurality of period value maps and selecting the period value map which best matches the resource availability request.

40 Preferably, the method of outputting generates data relating to the period value maps processed and to output a signal representing this data. The method of display may generate a table in dependence on this signal.

According to a further aspect of the invention there is provided a method of modifying a time-availability query, the method comprising:

45 receiving a first time-availability query;

receiving a resource selection template; and

generating a second time-availability query by applying the resource selection template to the first time-availability query.

50 Preferably, the method further comprises a method of further processing the query. Further processing may comprise evaluating the resource availability in accordance with the query.

Preferably, the query may be further modified by applying further resource selection templates, each successive modification being evaluated in turn.

55 In the present invention, in any aspect, a panel may comprise a simple digital display, such as an LCD or plasma screen as known in the art. The panel may be 'dumb' in that it has no or only rudimentary processing power of its own. Alternatively, the panel may be 'smart', in that it comprises a processing unit.

5 The digital content, for example the advertising data itself, comprising stationary and/or moving images or other multimedia, may be generated separately by a third party unconnected with the management of the time-availability system. This content may be provided by streaming to the panels in real-time for immediate display (which may involve temporary storage in a buffer), or downloaded in its entirety (for example overnight) for display at a later time. Such downloads may be pre-scheduled or made on-demand.

10 The data downloads may also comprise a panel schedule, specifying or used for determining the timings at which the content is to be displayed.

15 While the invention as presented in the particular embodiment described below is for the management of digital displays, the skilled person will appreciate that it has application to the management of any resource the allocation of which is time-based. For example, alternative embodiments may be used in the allocation of hospital beds, prison cells, cars for hire etc.

20 The invention also provides a computer program and a computer program product for carrying out any of the methods described herein and/or for embodying any of the apparatus features described herein, and a computer readable medium having stored thereon a program for carrying out any of the methods described herein and/or for embodying any of the apparatus features described herein.

25 The invention also provides a signal embodying a computer program for carrying out any of the methods described herein and/or for embodying any of the apparatus features described herein, a method of transmitting such a signal, and a computer product having an operating system which supports a computer program for carrying out any of the methods described herein and/or for embodying any of the apparatus features described herein.

30 The invention extends to methods and/or apparatus substantially as herein described with reference to the accompanying drawings.

Any apparatus feature as described herein may also be provided as a method feature, and vice versa. As used herein, means plus function features may be expressed alternatively in terms of their corresponding structure, such as a suitably programmed processor and associated memory.

35 Any feature in one aspect of the invention may be applied to other aspects of the invention, in any appropriate combination. In particular, method aspects may be applied to apparatus aspects, and vice versa.

40 Furthermore, features implemented in hardware may generally be implemented in software, and vice versa. Any reference to software and hardware features herein should be construed accordingly.

45 These and other aspects of the present invention will become apparent from the following exemplary embodiments that are described with reference to the following figures in which:

Figure 1 is a diagrammatic overview of a resource management / time-availability system;

Figure 2 is a block diagram of the main components of the time-availability server of Figure 1;

50 Figure 3 is a block diagram of the main components of the Availability Engine of Figure 2;

Figure 4 is a schematic representation of a resource availability bitmap;

55 Figure 5 is a schematic showing the data representation of a resource availability bitmap;

Figure 6 illustrates a method by which the Availability Engine constructs a resource availability bitmap;

60 Figure 7 illustrates an alternative method by which the Availability Engine constructs a resource availability bitmap;

Figures 8 illustrate the process of data scrubbing;

Figure 9 shows the query generation interface;

5

Figure 10 illustrates the availability querying process;

Figure 11 shows the structures of the matching period maps;

10

Figure 12 shows a hashing method;

Figure 13 shows the Availability Preview feature;

15

Figures 14 show some examples of the Availability Preview pie-clock of Figure13;

Figures 15 illustrate the pop-up tooltip and pop-up menu features of the pie-clock of Figure13;

Figure 16 shows the Zone Detail window of the Availability Preview;

20

Figure 17 shows an enhanced version of the query generation interface with the offset feature; and

Figure 18 shows the effect of the offset feature.

25

Overview

Figure 1 is a diagrammatic overview of a resource management / time-availability system 1 adapted to facilitate the management of advertisements on digital display panels 8,10. Digital content 2, such as advertisements (for example jpeg images or movie clips) is provided by a content provision server 4 to a content distribution server 6 for distribution to digital panels 8,10, according to availability and scheduling information provided by time-availability server 12. Time-availability system 1 is envisaged to work with a booking system 17 which enables panels to be booked by clients.

30

35

The digital panels comprise one or more digital display units, such as LCD or plasma screens, located at positions of high visibility to the advertisement target audience. Two different types of display panel are used: 'dumb' panels 8, which are essentially video display screens without additional processing power, and 'smart' panels 10, which further comprise a processing unit 11.

40

A dumb panel 8 can display content 2 as it is downloaded or provided by streaming (dumb panel 8 is provided with a simple buffer to smooth out interruptions due to network delays). Processing unit 11 enables a smart panel 10 to download content 2 at pre-determined times (for example, overnight) and store the content 2 for display at later times according to a downloaded panel schedule 13.

45

Time-availability server 12 maintains information on all the display panels 8,10 regarding their individual characteristics and also manages their allocation to display tasks.

50

Time-availability server 12 also provides scheduling information 14 to the content distribution server 6, which then in turn forwards appropriate scheduling information 13 along with the digital content 2 to the 'smart' panels 10.

55

It will be appreciated by those skilled in the art that the content provision server 4, content distribution server 6 and the time-availability server 12 need not be owned or operated by the same organisation. For example, time-availability server 12 may be operated by an advertising company which outsources the creative task of generating and providing content 2 to a third party (such as a graphics design company) which therefore operates the content provision server 4. Likewise, the advertising company may outsource the content distribution server 6 to an internet service provider (ISP).

60

Aspects of the time-availability server 12 will now be described in detail.

Time-Availability Server

Figure 2 is a block diagram showing the time-availability server 12 which is a computing platform comprising processor 20, non-volatile read-only memory ROM 22, and volatile random access memory RAM 24. Further storage is provided by disk storage 26. A user interacts with the time-availability server 12 via standard computer input means such as the keyboard 28 and mouse 30, and receives output via the display 32. Such a computing platform is known to those skilled in the art and is provided by, for example, Sun Microsystems, Inc., running a UNIX operating system 40, such as Solaris.

Figure 2 shows that when the time-availability server 12 is operating, the Availability Engine 34 software application is loaded into RAM 24; resource database 36 is stored in disk storage 26.

The Availability Engine 34 is discussed in more detail in the following section.

Resource database 36 is a relational database, such as a SQL database, and stores details of all the digital panels 8,10. Each panel 8,10 is identified by a unique panel ID. The panel ID is a number allocated in sequence starting at 1 for the first panel, and incrementing by 1 for each new panel added to the system. The panel ID is stored in 32-bit entities, thereby allowing the system to accommodate up to approximately 4 billion ($= 2^{32}$) panels. A typical UK-wide system as currently in use may have upwards of 150,000 panels. Each panel possesses one or more of the following characteristics: panel size, panel type (including model number, whether 'dumb' or 'smart'), location, orientation (portrait or landscape), aspect ratio, whether designated as part of an advertising campaign (and if so, which one), asset number, contract number, whether operational or not (if damaged or obscured, for example), whether for example the panel is participating in Postar advertising audience research trials etc. Resource database 36 also holds the full scheduling details for each panel i.e. information as to whether the panel is available or unavailable for booking at a particular time.

Availability Engine

Figure 3 is a block diagram of the Availability Engine 34 showing its main components. The Availability Engine 34 software application is written in the C programming language.

In overview, Availability Engine 34 extracts relevant panel availability data from the resource database 36 and uses it to construct one or more resource availability bitmaps (discussed below). These are essentially 2D data arrays representing panel availability during discrete time slots. Typically, there exist a number of resource availability bitmaps at any one time, each representing the availability of a group of panels, the group membership being defined by pre-defined selection templates. Availability Engine 34 uses resource availability bitmaps to determine resource availability in response to resource availability queries in preference to directly querying the resource database 36.

User interaction with the Availability Engine 34 via I/O module 64 is by means of a web front-end, written in Java. In this embodiment, the web front-end runs as a stand-alone application, although in alternative embodiments it may run within a common web browser, such as Microsoft Internet Explorer or Mozilla Firefox.

Details of the other components of the Availability Engine 34 are described below in relation to the following operations:

1. Generation of resource availability bitmap(s)
2. Resource (pre-) selection templates
3. Memory scrubbing
4. Availability querying
 - Query generation
 - Query processing
 - Pattern matching
 - Filtering
5. Availability Preview

1. Generation of resource availability bitmap(s)

Figure 4 shows a schematic representation of how each resource availability bitmap 70 comprises a 2D array. Availability is tracked in time slots, with each slot represented by a binary element which is in one of two states: 1 indicating the panel is available during that time slot; 0, indicating the panel is not available.

Each column 90 of the array is associated with a particular panel, referenced by the unique panel ID p_n (i.e. each panel ID is used as an index to the bitmap), with successive elements indicating the availability of the panel from time 1 to time t_N ; each row of the array therefore corresponds to the availability of panels at a particular time, delimited by $startPeriod_N$ and $endPeriod_N$.

The overall size of a resource availability bitmap 70 is determined by three factors:

- the word size of the computing architecture used
- the number of panels for which availability needs to be simultaneously determined i.e. the number of panels belonging to the panel group
- the number and granularity of the time-availability divisions

The width of the resource availability bitmap 70 (in terms of column size and number) is determined by the first and second factors. In this embodiment, the computing architecture uses a 64-bit word length and therefore the width of the resource availability bitmap 70 is a multiple of 64-bit entities. For example, to accommodate up to 3000 panels the resource availability bitmap 70 comprises 48 * 64-bit words (= 3072 bits).

The number of rows in the resource availability bitmap 70 is determined by the third factor: the number and granularity of the time-availability divisions. Availability information is held for a four-year time period, comprising previous, current, next and next-plus-one years. At the close of each year, a year-end process time-shifts the data by one year accordingly, replacing the information for what was the 'previous' year by the information for what was the 'current' year, replacing the information for what was the 'current' year by the information for what was the 'next' year and replacing the information for what was the 'next' year by the information for what was the 'next-plus-one' year, thereby freeing up the space allocated for the 'next-plus-one' year. A 4-year time period corresponds to 1462 days (= (365 * 3) + 366), as every 4-year period usually contains a leap year of 366 days (barring exceptions such as 2100, which is unlikely to be of present concern). For a minimum time period of 5 seconds, this results in each day having (60 * 60 * 24) / 5 = 17,280 periods allocated. For the 4-year period this equates to 17280 * 1462 = 25,263,360 periods in total.

Thus, for a test sample of 3000 panels, the size of the resource availability bitmap 70 used to represent the availability the panels in the test data over a four year period in 5 second intervals is some 3072 columns by 25,263,360 rows, which requires some 9 GB of memory.

Figure 5 shows how the resource availability bitmap 70 is stored as a sequence of record elements 94 (or 'period maps'). These are created and initialized at process start-up, and each corresponds to a successive row of the resource availability bitmap 70, and has the following structure (presented here in pseudocode which will be readily understood by those familiar with the C programming language; the /* and */ delimiters signify comments):

```

typedef struct {
    mutex_t *periodLock    /* Period lock */
    BMAP *map;             /* Panel Map */
    int startPeriod;       /* Start period */
    int endPeriod;         /* End period */
    short blockId;         /* Memory block allocated from */
} PMAP;
    
```

wherein the elements of each period map are as follows:

periodLock - used to synchronize updates to the period map. Only one process at a time is allowed to update the period map.

map - where the panel bit map is stored, e.g. a pointer to a memory address

5 *startPeriod* - used to store the start period. This is a value between 1 and 25263360. This is a value that represents a period within the scope of the system. Currently period 1 represents the start of the 4-year period beginning at 01/01/2007 00:00:00

10 *endPeriod* - used to store the end period. This has a similar format to the startPeriod described above i.e. a value of 25263360 represents the end of the 4-year period at 31/12/2010 23:59:59

15 *blockId* - used to store the memory block the memory map element is allocated from.

When updating (or otherwise making changes to) the database with new availability information (for example, as panels are booked), it will be important to ensure that suitable data locking methodology is observed to prevent update clashes occurring. This facility is provided for by the periodLock element mentioned above. This is essentially what is known in the art as a mutual-exclusion or mutex lock. Whenever a process accesses a period map, it first checks to see whether the periodLock element associated with that period map is set. If the periodLock element is set, the process is not permitted to modify the period map at that time. The process must then wait until the periodLock element is cleared (i.e. unset). If the periodLock element is not set, the process is free to modify the period map – and it sets the periodLock element to gain exclusive control of the period map while it does so, thereby excluding other processes attempting to modify the period map for the duration. The process clears the periodLock element, relinquishing control, once it has completed its modifications. This ensures updates behave in an atomistic fashion, that is, with an “all-or-nothing” allocation, whereby in the case that two users or processes are seeking to update the availability information for the same panels in the resource database 36, one user or process (selected randomly, or else according to some client metric such as desirability) is met with complete success and the other has all changes rejected (and a suitable error message generated as required).

35 A second type of resource availability bitmap is used to represent resources on which an ‘option’ has been taken i.e. resources which are provisionally booked.

A third type of resource availability bitmap is used to represent the ‘potential’ availability i.e. if all panels were available.

40 Those skilled in the art will appreciate that the ‘options’ and ‘potential’ bitmaps are not essential in all embodiments. Furthermore, similar functionality is possible by means of using values other than binary in a single resource availability bitmap – although this requires a corresponding increase in complexity in some of the underlying logic operations.

45 A transaction process maintains the availability information held in in resource database 36 is in synch with that in resource availability bitmap 70.

2. Resource (pre-) selection templates

50 As discussed above, depending on the number of panels and the number of time-availability divisions (representing both the length of the time period under consideration and its granularity), the size of the resource availability bitmap 70 can become considerable. It will be appreciated by those skilled in the art that it is common to have advertising targeted to particular audiences, locations or types of panels (for example, restricted to those panels which can support a particular type of media format). Therefore in practice only a subset of all panels will typically need to be considered, and consequently substantial gains in computational efficiency can be made by considering only a subset of panels, as represented by one or more ‘reduced’ resource availability bitmaps.

Panels may be pre-selected according to specific requirements. For example, one group of panels may comprise all panels at a particular location A; another group of panels may comprise all panels of at least dimensions X x Y; further panel groupings may be defined from any panel attribute (or combination thereof) stored in the resource database 36. Resource selection templates defining the membership credentials for each group are stored long-term in disk storage 26 and loaded into RAM 24 as required.

Figure 6 illustrates how a reduced resource availability bitmaps, one for each panel group, are generated from a 'master' bitmap 70-1, comprising availability information for the entire complement of resources. Resource selector 50 obtains group information (or group membership information) 73 from resource template library 38 and uses it to construct corresponding selection filters 74. Boolean operation OP1 (typically an AND or OR operation) is used to apply individual selection filter 74-1 (essentially a linear bitmap) to master bitmap 70-1, thereby generating reduced bitmap 70-2 corresponding to the required group. Columns corresponding to unselected resources may be deleted. Individual selection filters 74, such as 74-2 and 74-3, may themselves be combined in one or more Boolean operations OP2 to form composite selection filters for more complex selections. Resulting reduced bitmap 70-2 is stored in bitmap library 54 for future recall.

Figure 7 illustrates an alternative method of generating reduced resource availability bitmaps, wherein resource selector 50 obtains resource selection template information 73 from resource template library 38 and uses it to construct corresponding selection queries 72 (for example, SQL queries). These selection queries 72 are then used to select corresponding availability information from resource database 36 and bitmap generator 52 constructs corresponding reduced bitmaps 70-3 which are stored in bitmap library 54 for future recall.

3. Memory scrubbing

As mentioned above, to represent the availability of a single panel in five-second slots during a single day will require approximately 17,000 binary elements. Tracking panel availability in five-second slots for the four year time period will therefore require approximately 25 million binary elements per panel. This can become an issue as the number of panels increases. For example, a test sample set of 3072 panels would therefore require at least 9 GB ($= 3072 * 25,263,360 / (8 * 1024^3)$) of memory merely to hold the availability data. A more realistic data set for 150,000 panels, as found in some large-scale commercial advertising organisations, would require over 440 GB of memory. Furthermore, for reasons of processing speed this data is preferably held in fast RAM 24 rather than comparatively much slower disk storage 26, making suitable computing systems expensive.

In order to reduce the computational burden the resource availability bitmaps 70 stored in the bitmap library 54 are optimised by means of a 'memory scrubbing' process performed by the memory scrubber module 58 shown in Figures 6 and 7. This runs a housekeeping process every minute which, for each bitmap 70, iterates through the period maps, in period order (i.e. effectively through successive rows of the resource availability bitmap array), and rationalizes them accordingly, freeing up memory.

In particular, memory scrubber 58 seeks patterns within the availability data including contiguous slots of availability (or non-availability) which may be combined and therefore described as one single, larger slot.

Figures 8 show the workings of the memory scrubbing process. As initially configured, each time period or slot is represented by a period map, corresponding to a row in the resource availability bitmap 70. If, for any two consecutive period maps 102,104 (representing contiguous time periods), the panel bitmap elements are identical (i.e. the same panels are available for both time periods), then in principle a single period map 106, with suitable startPeriod and endPeriod values, can represent both time periods. In the limit, if the panel bitmaps are identical for all time periods, all time periods can be represented by a single period map.

Memory scrubber module 58 therefore compares the panel bit map element of the current period map with that of the next period map. If the panel bitmaps are an exact match, the endPeriod element of the current period map is set equal to the value of the endPeriod element of the next

(current + 1) period map and the next period map is then disposed of. This is shown in the following pseudocode:

```

5       if periodMap[current]->map == periodMap[current + 1]->map {
           periodMap[current]->endperiod = periodMap[current + 1]->endPeriod
           destroyPeriodMap(periodMap[current + 1])
       }

```

10 The decision as to whether to merge two period maps, say PMAP₁{startPeriod₁; endPeriod₁; BMAP₁} with PMAP₂{startPeriod₂; endPeriod₂; BMAP₂} is taken according to whether the following conditions are met:

- o The period maps have time periods or slots which are adjacent
i.e. startPeriod₂ = endPeriod₁ + 1
- o The period maps have identical panel availabilities i.e. BMAP₁ = BMAP₂

15 If all these conditions are met, the slots are merged or concatenated. In the limit, if all period maps had identical panel bitmaps, the memory scrubbing process would merge them into a single period map.

20 As changes in availability occur, the number of period maps therefore increases and decreases accordingly to accommodate and reflect the complexity of the underlying availability.

4. Availability querying

25 Availability querying is expected to take place in an interactive session with, for example, sales staff negotiating with clients in real-time in a telephone conversation. The emphasis is therefore on the speed at which the availability of panels (and groups thereof) can be determined.

30 Time-Availability System 1 is designed to allow the user to easily query the availability of a pre-determined group of panels by means of an intuitive user interface (UI). It is particularly important that the user interface is both clear and simple to use as it is required to impart a considerable amount of detailed information which needs to be rapidly assimilated (as in an interactive telesales scenario), often by staff who may not have had the benefit of experience gained through querying databases by more traditional means. Time-Availability System 1 therefore uses distinct user interfaces to assist in the generation of appropriate availability queries and to facilitate the interactive investigation of different scenarios in real-time.

Query Generation

The typical form of an availability query request will comprise the following elements:

- target number of panels i.e. at least N panels
- duration of time period for which panels are to be available i.e. T seconds, typically up to 60 seconds
- time period repetition rate i.e. every R minutes
- time period interval i.e. availability to be between start S and end E on date D, typically < 24 hours

45 Hence an availability query will therefore take the form of a set of parameters {N, T, R, S, E, D}. Multiple start S and end E time period window delimiters may be defined.

50 Figure 9 shows the query generation interface (or 'Select Date' window) 120 for communicating with the time-availability system 1. The query generation interface 120 comprises three main elements:

- o date-request interface 122, located on the left-hand side of the window;
- o repetition-request interface 124, located in the upper right-hand side of the window;
- and
- o time-request interface 126, located on the lower right-hand side of the window.

60 Standard input window operation buttons (OK, Cancel, Reset and Clear) are provided; these will be familiar to those skilled in the art.

▪ **date-request**

The date-request interface 122 comprises a month-to-view calendar representation. Individual dates (D) 130 and ranges thereof are selectable by mouse-click in the conventional way. In use, the user first selects the required month to view by means of drop-down menu 131.

The advertisement industry may work on a month-by-month basis, having 12 'cycles' per year (one cycle corresponding to each calendar month) with successive advertising 'weeks' starting on the 1st, 8th, 15th and 22nd day of the month and ending on the 7th, 14th, 21st and the last day of the month respectively. More recently, the industry has moved to having 13 cycles per year, each of 28 days, with the advertising 'week' running from Monday to Sunday. Date-request interface 122 allows for straightforward translation between the systems as required.

The style of the calendar is determined according to a client-specific configuration. Three different formats are supported, which are defined by the date-type variable as:

"IM"	the traditional cycle of months, therefore a cycle period represents a week in most cases, but the final week of a month always extending to the last day of the month;
"28"	representing a standard, 7-day week cycle (Monday to Sunday);
"IV"	a Variable cycle, based on days i.e. a selection of 4 represents 4 days.

For example, where a client contract specifies (by means of an initial configuration screen, not shown) the traditional "IM" cycle is to be used, the date-request interface 122 will present a calendar with the appropriate key dates highlighted as shown in Figure 9 i.e. the calendar is displayed with each week starting with Monday, with the traditional 'start' days (1st, 8th, 15th and 22nd) highlighted. The user has selected start date S, Friday 1 February 2008, and the number of cycles (=2), thereby determining the end date E, Thursday 14 February 2008.

An alternative query generation interface (or 'Select Date' window) 120 is provided with an enhanced date-request interface 122 which allows for the setting of the date-type variable directly. This interface is used for ad hoc availability queries.

The user selects the 'Cycle Size' (in periods) via 'Cycle Size' input box 132. Cycle Size is determined by the date type, "IM", "28" and "IV".

Alternative embodiments allow for the selection of specific days, either by day or individually by date.

For example, in one alternative embodiment, to select the day, a further panel is provided in the dialog showing checkboxes with the following labels: *Every Day, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday and Sunday*. The default is for *Every Day* to be checked. This allows for specific days only within a selected date sequence to be selected, allowing the user to perform a single query, whereas in other embodiments multiple queries would be required if specific days only are required.

An alternative embodiment to allow selection of date uses, for example, a 'push-button' display, the individual days being directly selected by means of the user 'clicking' on the individual dates (D) 130.

These day/date selection alternatives may be used individually or in combination.

These selection methods signify a marked departure from the traditional 'cycle' format as currently used in the art.

The time period (both dates and times) under consideration is displayed by the time period indicator 134.

▪ **repetition request**

The repetition request interface 124 allows the user to specify the desired display repeat frequency of the advertisement on the panels by means of entering values in the appropriate fields.

The user may set the time (T) for which the advertisements are to be displayed (for example, 5 seconds) in the 'Duration' field 136; how often this display duration is to be repeated (R), for example, every minute (i.e. in minute intervals), in the 'Every' field 138 (this feature is also known as the 'interval' feature); and the desired number of panels (N) on which the advertisement is to be displayed (for example, 1801) in the 'Target' field 140. To assist in selecting a suitable target value, the repetition request interface 124 displays the total number of panels available.

▪ **time-request**

The time-request interface 126 allows the user to select the time of day interval during which the advertisement is to be displayed. The day is presented as a 24-hour clock face, divided into one-hour major divisions which are marked on the circumference. In this embodiment, each hour is further sub-divided into four; that is to say, the smallest selectable unit of time is fifteen-minutes. Other embodiments may have finer or coarser subdivisions. Extending from the centre of the clock face to the circumference are zone defining radii 142. These can be selected and manipulated by the user (via conventional click and drag mouse operations) to sweep out a sector of the clock face and therefore define an active zone corresponding to the time of day during which the advertisement is to be displayed (at the repetition rate as defined by means of the repetition request interface 124 described above).

The leading edge 142-1 of the active zone corresponds to the start time (S) of the period during which the advertisement is to be shown, and is marked by a green radial line; the trailing edge 142-2 of the active zone corresponds to the end time (E) of the period during which the advertisement is to be shown, and is marked by a red radial line.

Multiple active zones may be defined within a day.

Hence, by using the query generation interface 120 described above, the user can easily formulate a date-time-repetition request for the display of advertisements in an advertisement campaign – comprising parameters {N, T, R, S, E, D} as mentioned previously.

In the above example, when requesting the target number of panels, N, it was not specified which individual panels were to be selected. Clearly, this would be a needlessly troublesome requirement when all that is required is for a certain total number of panels to be targeted. However, a more advanced feature allows for the user to specify target panels to the query generation interface 120 by a selection template, as described above. For example, a standard windows user interface facility, as will be familiar to those skilled in the art, allows for a selection template to be selected by the user from a list and, by using a mouse to drag-and-drop, deposited on the query generation interface 120, indicating to the query generation interface 120 that the target panels comprise those panels referred to by the selection template.

Multiple selection templates may be applied simultaneously for more complex selections - resulting in one or more Boolean operations between selection filters, as described previously in relation to operation OP2 of Figure 6. Thus the user can opt for either requesting a target number of (at least) N panels or specifying a particular set of panels (to represent this possibility, the symbol N* is used in the following).

Use of selection templates is indicated by a suitable indicator on the generation interface 120.

A 'Preview' check box 144 is provided which, when checked, submits the availability query to the Availability Engine 34 for processing (and allows for the resulting panel availability information to be viewed).

Query Processing

Figure 10 illustrates the process by which the Availability Engine 34 processes availability queries. According to the panel group requested (for example, by the drag-and-drop of a selection template onto query generation interface 120, as described above), resource selector 50 selects the appropriate resource availability bitmap 70 from the bitmap library 54. User interaction via the user interface (described in detail below) and the I/O module 64 results in the

availability query generator 56 generating an availability query 78, comprising parameters {N*, T, R, S, E, D} as mentioned previously.

The process of matching the availability query 78 to the resource availability (as represented by the resource availability bitmap 70) is undertaken by the pattern matcher 60 and filter 62 in an iterative process by means of building up a succession of 'matching period maps' 80. These are transient maps used to represent the periods and panel maps that match the supplied criteria. During processing, matching period maps 80 are stored in memory in matching period map store 59.

Figure 11(i) shows a matching period map 80 with the following structure:

```

typedef struct {
    int start_period;           /* Start period for map */
    int end_period;           /* End period for map */
    int slot_size;           /* Slot size in seconds */
    int status;              /* Status = 0 or 1 */
    int objectType;         /* Object Type = 1, 2 or 3 */
    int last_merge_start_period; /* Last merged Start period for map */
    BMAP *map;              /* Panel bitmap */
} MAPMAP;
    
```

wherein the elements of each matching period map are as follows:

- start_period* - used to store the matching start period
- end_period* - used to store the matching end period
- slot_size* - the size in seconds of the slot, calculated by subtracting the seconds element of the *start_period* from the seconds element of the *end_period*. Slot sizes are calculated by modular arithmetic:
 To isolate the day element :
 $day = period - (period \% common_periods_per_day)$
 To isolate the time element:
 $time = period \% common_periods_per_day$
 The slot size in seconds is calculated as follows:
 $slot_size = ((endPeriod \% (60 / common_period_unit_in_seconds)) - (startPeriod \% (60 / common_period_unit_in_seconds)) + 1) * common_period_unit_in_seconds$
 In the present configuration :
 $common_periods_per_day = 17280$
 $common_period_unit_in_seconds = 5$
- status* - a flag to indicate whether this particular matching period map is still required, a value of 0 means it is, a value of 1 means it should be discarded
- objectType* - used to indicate the type of bitmap the matching period map represents, where a value of 1 = 'availability', 2 = 'potential' 3 'option' map.
- last_merge_start_period* - indicates the last start period the period map was merged with. This assists with the pattern-matching process by enabling patterns to be more easily identified, as explained below.
- map* - is the panel bitmap.

These matching period maps 80 are used in the pattern matching and filtering processes as described below.

▪ **pattern matching**

The pattern matching process attempts to match the requirements of the availability query 78 to the resource availability (as represented by the resource availability bitmap 70).

In essence, pattern matcher 60 creates a plurality of matching period maps 80, each of which represents a different attempt by the Availability Engine 34 to match the date-time-repetition request submitted by the user to the availability indicated by the appropriate resource availability bitmap 70.

- use of a recursive method

The pattern matching process begins with the requested start time period, S, and traverses the period maps in order until it reaches the requested end time period E. For each intervening time period, the matching process compares the selection template with the panel map BMAP of the corresponding period map 94 for that time period. Those matching period maps 80 which form a match (according to preset requirements, specific to the art), with the period map 94 under consideration are built up accordingly (i.e. the slot size increases, a new end period is set etc) and stored as an array in memory.

In the field of advertising, interest is in regular patterns of availability such as, for example, a specific number of panels for a certain duration, at the same time in successive days over a period of weeks. The specific matching requirements (which may be determined by the specific system installation, configured centrally or selected by the user via a further option provided by the query generation interface 120) therefore typically require matching contiguous periods of equal slot size and identical BMAPs on the same day, but may also, as in the presently described embodiment, involve matching non-contiguous periods, either within the same day or across days (for example, similar time periods on subsequent days).

The process of pattern matching is instigated each time a match is found between the bitmap supplied (i.e. the selection template) and the matching period maps 80.

Panel IDs are passed in as strings in delimited form, and are expanded into a bitmap in the exact same format and size as those in the period maps. For example, the string "1 | 1000 | 1132", wherein the vertical bar is a delimiter, sets the first, 1000th and 1132nd bits of the selection template, corresponding to selecting the first, 1000th and 1132nd panels.

The repetition request is handled in the following way. For example, for a time-availability request requesting a display of 10 seconds duration at an interval of 1 minute, referring to Figure 9, the display period is the 'duration' (T) as set in the 'Duration' field 136; the repetition interval is the time gap between each display (R) as set in the 'Every' field 138, i.e.:

```

slot_size = 5           // 5 second slot size
duration = 10 / slot_size // divide by slot size
interval = (1 x 60) / slot_size

```

Hence, in this example, for a 5 second slot size, the required duration T (10 seconds) is two slots long, repeated every R (1 minute) = 12 slots. The following pseudocode shows how the availability checking is done:

```

period = startDateTime
while period <= endDateTime {
  for index = 0; index < duration; index++ {
    result = AND(availability[period + index]->map, queryMap)
  }
  period = period + interval
}

```

The result of ANDing the supplied bitmap (i.e. the selection template) with the map element of the period map is tested to see if any bits remain set (indicating that at least one of the requested panels is available). If any bits remain set (i.e. there is at least one panel available) the start and

end periods together with the resultant bitmap are compared to the array of matching period maps found so far. For each matching period map the newly found start and end periods are processed using the following pseudo code:

```

5  if day(mMap[current]->start_period) == day(new_start)
    AND day(mMap[current]->end_period) == day(new_end){
                                     /* contiguous time periods */
10  if isContiguousTime((mMap[current]->end_period, new_start){
    if (mMap[current]->slot_size == new_slot_size {
                                     /* same slot size */
        if mMap[current]->map == new_map {
                                     /* same BMAPs */
            mMap[current]->end_period = new_end
            mMap[current]->last_merge_start_period = new_start
            mMap[current]->slot_size =
15  getSlotSize(mMap[current] ->start_period, mMap[current] ->end_period)
        }
    }
20  } else {
                                     /* next minute */
    if (mMap[current]->start_period + oneMinute) == new_start
        AND (mMap[current]->end_period
25  + oneMinute) == new_end {
        if (mMap[current]->slot_size == new_slot_size {
            if mMap[current]->map == new_map {
                mMap[current]->end_period = new_end
                mMap[current]->last_merge_start_period = new_start
                mMap[current]->slot_size =
30  getSlotSize(mMap[current] ->start_period, mMap[current]->end_period)
            }
        }
    }
}
} else {
                                     /* next day */
35  if (day(mMap[current]->last_merged_start_period) + oneDay) == day(new_start)
    AND (day(mMap[current]->end_period) + oneDay) == day(new_end) {
        if time(day(mMap[current]->start_period)
40  == time(new_start)
        AND time(mMap[current]->end_period) == time(new_end) {
                                     /* same time of day */
            if (mMap[current]->slot_size == new_slot_size {
                if mMap[current]->map == new_map {
                    mMap[current]->end_period = new_end
                    mMap[current]->last_merge_start_period = new_start
45  mMap[current]->slot_size =
                    getSlotSize(mMap[current] ->start_period, mMap[current]->end_period)
                }
            }
        }
50  }
}
}

```

Thus only those matching period maps 80 which satisfy the specific pattern-matching criteria are built up. As those skilled in the art will appreciate, the above pseudo code represents one set of specific matching criteria; other criteria may also be used.

Each time a match is found the method calls itself again with the start, end periods and the bitmap of the newly merged matching period map. This continues recursively until no further matches are found.

60

As will be appreciated, the pattern-matching process may take many different forms according to the specific requirements for particular patterns to be considered 'matched' or 'contiguous'. The *last_merge_start_period* field assists with this.

- 5 Consider the following scenario, wherein the start time S is 1 May 2008 10:00:00 (for clarity, only the start times are shown; the slot sizes and BMAP values are presumed to match):

Matching Period Map	<i>startDateTime</i>
A	1 May 2008 10:00:00
B	2 May 2008 10:00:00

- 10 For matching according to start times, the matching period maps A and B, although on consecutive days, will be considered to be contiguous and are therefore merged.

Consider now a third matching period map, C:

C	3 May 2008 10:00:00
---	---------------------

- 15 The comparison test for contiguity of C with A and B is not based on the start time S (which corresponds to the start time of A) but on the *last_merge_start_period* i.e. the start time of B. Thus the same periodicity test can be applied for C as was applied for B, and C can therefore likewise be merged.

- 20 The pattern matching process determines if the requirement can be met with available panels. This allows for automatic accommodation of multiple client sequence requests which would otherwise only be possible by deliberate manual allocation. Where the availability engine determines that the interval request cannot be satisfied, the user is informed and the time-request interface 126 is presented for an alternative request.

25

- use of a hashing method

The pattern-matching process as described above relies on a recursive method wherein each new matching period map 80 has to be compared with each of the matching period maps already discovered. Thus for example, if 1000 matching period maps have already been discovered the next matching period map discovered will need to undergo 1000 comparisons or contiguity tests. As the number of time periods under consideration increases (for example, when the period under consideration is for a nine-month advertising campaign rather than a two-week cycle), the time taken to complete the pattern-matching task can become excessive with results no longer available in real-time.

35

An approach to speeding up the pattern matching involves reducing the size of the data set by pre-selecting matching period maps before matching.

- 40 The pattern matching process can therefore be assisted by means of a hashing method. This complements the pattern-matching by reducing the number of comparisons or contiguity tests required. The pre-selection of matching period maps before matching allows for a table look-up and attempted matching to a subset comprising potentially suitable matching period maps rather than a recursive iteration over all the matching period maps already discovered.

- 45 Figure 12 illustrates the effect of the hashing method. In essence, the existing found matching period maps 80 are assigned by means of a hashing function to 'bins', for example, according to their start times; for a contiguity match according to start times the new matching period map therefore only needs to be compared to those other matching period maps in the same starting time 'bin' rather than to all previously discovered matching period maps.

50

In practice, the hashing method assigns the found matching period maps 80 into a table 92 indexed by a parameter, such as the start time, which can be easily isolated or derived from the matching period map. Isolation of the time element of the start date time of the matching period map can be achieved by, for example, the following pseudo code:

55

$$time = period \% periods_per_day$$

where % is the modulo operator and *periods_per_day* is the number of slots supported per day i.e. 86400 (seconds in a day) divided by the slot size in seconds. Where the minimum slot size is 5 seconds, this algorithm would provide a value between 1 and 17280.

This value is used as an index into an array of matching period maps which are chained together by means of a memory pointer, which provides the address of the next matching period map. To facilitate this, the structure of the matching period maps 80 described previously is extended with an additional field, *next_match_map*, as follows and as shown in Figure 11(ii):

```

typedef struct {
    int start_period;
    int end_period;
    int slot_size;
    int status;
    int objectType;
    int last_merge_start_period;
    next_match_map; /* chained matching period map */
    BMAP *map;
} MAPMAP;
    
```

Figure 12 shows matching period maps 80 A, B and C being assigned by a hashing function to a table 92. In this example, each matching period map 80 A, B and C has the same parameter (such as a start time), k, and is therefore being assigned by the hashing function to the same 'bin' of a table 92, indexed as 'k*' (k* rather than k to indicate that the index need not be the same as the parameter, only related to it; likewise, A*, B* and C* to indicate that a similar principle applies to the matching period maps in that their content need not be copied into table 92, only a reference, such as a pointer to the corresponding memory address, need be provided). Also shown are matching period maps G and H, of common parameter m, which were earlier assigned by the hashing function to the 'bin' of index 'm*' as G* and H*. To match to those matching period maps with a start time of k it is therefore only necessary for the period matching process to match to the pre-selected matching period maps A, B and C, rather than having to match (recursively) with all matching period maps.

By traversing the chained period maps (by means of the means of the memory pointer, which provides the address of the next matching period map) a new matching period can be tested to determine if it matches the merging criteria as described previously (for example, matching contiguous periods of equal slot size and identical BMAPs on the same day, or alternatively, matching non-contiguous periods, either within the same day or across days, as in similar time periods on subsequent days); if it fails a new matching period map is created at the end of the chain.

For example, consider the following scenario:

Matching Period Map	startDateTime	next_match_map
E	1 May 2008 10:00:00	F
F	13 May 2008 10:00:00	-
G	15 May 2008 10:00:00	-

Here, if matching period maps E and F are considered to be contiguous and are merged, G does not match the merging criteria (the separation in days between E and F is not the same as that between F and G). Instead, a new matching period map F* is created - and G is chained to it.

Matching Period Map	startDateTime	next_match_map
E	1 May 2008 10:00:00	F
F	13 May 2008 10:00:00	-
F*	13 May 2008 10:00:00	G

G	15 May 2008 10:00:00	-
---	----------------------	---

Similar hashing schemes could be employed for day, hour and minute.

5 For example, in order to support a 4-year time period, corresponding to 1462 days (= (3 * 365) + 366), a hashing algorithm would use values between 1 and 1462. This is achieved by the following pseudo code:

day = period / periods_per_day

10 where / is the division operator.

By using a combination of these two hashing methods, it is possible to create an ordered (by day and time) list of matching period maps (to be used separately, jointly or in sequence, according to the form of availability query) without resorting to conventional sort methods which can be
15 intensive in both computational power and elapsed time.

The hashing method has been shown to improve time-availability determination performance by factors of up to forty times.

20 **▪ filtering**

As will be appreciated, the availability data for a panel is unlikely to be entirely random in nature but instead will exhibit some degree of structure. Exploiting this structure to optimise the resource availability bitmaps 70 by, for example, reducing their size and thereby easing the computational requirements has already been discussed with respect to resource pre-selection and the
25 operation of the memory scrubber module 58.

Such optimisation can also be done with a view to providing for the type of availability which is preferred in the art, thereby further enhancing the speed of the availability queries and enabling successful matches between client requirements and panel availability. For example, in the
30 present embodiment, it is preferable in the art to have long periods of availability, potentially with a determined or constant, if smaller, number of available panels, rather than fragmented periods with a variable number of available panels.

Referring back to Figure 10, filtering, as performed by the filter 62 of the Availability Engine 34 on
35 the matching period maps 80, is therefore used to determine the longest consecutive period of time the largest number of panels is available for.

The purpose of the filtering process is to try and establish matching period maps for the most contiguous periods, it is not concerned with the number of panels. This is referred to as "Non-split". This process works purely on the array of matching period maps after all other processing has completed. It is similar to the pattern matching process with one exception, it does not compare the *map* elements, it ANDs them together. The following pseudo code replaces the map comparison in the pattern matching pseudo code described previously:

```

45   new_map = bitmap_and (curr->map, mMap[i]->map)
      if bitmap_anysset (new_map) {
          mMap[i]->map = bitmap_exclude(mMap[i]->map, new_map)
          dummy = bitmap_exclude(curr->map, new_map)
          if bitmap_anysset (dummy) {
50             create_match_map (curr->start_period, curr->
                                     end_period, dummy)
          }
          if bitmap_anysset (mMap[i]->map) == 0 {
              mMap[i]->status = 1;           // no panels remaining discard
55       }
          curr->end_period = mMap[i]->end_period
          bit_map_copy (curr->map, current)
      }

```

The results of the filtering process may be appreciated from the following example:

Before Filter Process:

	Start	End	No. Panels	Slot size
Matching Period Map 1	11/02/2008 18:00:00	24/02/2008 23:59:09	2066	10 sec
Matching Period Map 2	11/02/2008 18:00:10	24/02/2008 23:59:59	3072	50 sec

5 As can be seen in the above table, the initial determination of matching period maps indicates that there are two matching period maps in the period 18:00:00 11/02/2008 to 23:59:59 24/02/2008, of which MPMMap2 offers the larger number of available panels (3072) for the larger slot size (50 seconds).

10 The filtering produces the following revised MPMMaps:

After Filter Process:

	Start	End	No. Panels	Slot size
Matching Period Map 1	11/02/2008 18:00:00	24/02/2008 23:59:59	2066	60 sec
Matching Period Map 2	11/02/2008 18:00:10	24/02/2008 23:59:59	1006	50 sec

15 As can be seen, filtering has re-allocated panels between the matching period maps and now shows that a revised MPMMap1 can actually accommodate a slot size of 60 seconds, albeit with a reduced number of 2066 panels.

5. Availability Preview

20 Once the pattern matcher 60 and filter 62 have completed their processing, the extent to which the overall (or specific) target request N* can be met is determined by comparing, for each time period over the time interval specified, the best-match matching period map 80-3 with the original request. Where only an overall target number of panels has been specified, the total number of panels marked as available in each consecutive time period in the best-match matching period map 80-3 is compared to the requested target number of panels. Where specific panels have
25 been requested, the corresponding selection template (or combination thereof) is compared in a Boolean operation (for example, an AND operation) with the corresponding panel bitmap of the best-match matching period map 80-3. The results, with respect to pre-determined thresholds, are displayed in the Availability Preview mode.

30 Thus the results displayed in the Availability Preview are essentially direct representations of the information stored in the best-match matching period map 80-3.

35 Figure 13 shows the Availability Preview window 150, as triggered by checking the 'Preview' check box 144 in the Select Date window 120, displaying the result of a test date-time-repetition request query on panel availability test data.

40 Availability Preview window 150 presents a 'pie-clock' indicator 152 comprising a 24-hour clock face as used in the time-request interface 126 described above, which indicates to the user, by means of the colour-coding, whether, and to what extent, the date-time-repetition request can be fulfilled.

45 Those skilled in the art will appreciate that the use of a pie-clock format is merely one of many alternatives; it is used here as it complements the time-request interface 126 of the query generation interface (or 'Select Date' window) 120, and therefore assists the user in understanding the relationship between the two. However, other formats may also be envisioned, for example, a linear or block form.

In the present embodiment, as has been described earlier in relation to the 'availability', 'options' and 'potential' bitmaps, the Time-Availability System 1 allows for three different categories of

panel availability. This is represented by the division of the pie-clock into three concentric rings, and which are further colour-coded by section according to how well the panel availability meets the date-time-repetition request. A colour-key 154 is provided for the user.

5 In order, from the innermost ring, the three panel categories are as follows:

▪ **Potential**

10 The first, innermost ring 156, corresponding to the processing of the 'potential' bitmap, represents the total inventory of panels. This therefore indicates the potential for meeting the date-time-repetition request, in that, clearly, if the date-time-repetition request for panels (the 'target') exceeds the total inventory of panels (Potential < target), the request cannot be met even if all of the panels are available. In that case, the innermost ring 156 is coloured light grey. If the target value does not exceed the total inventory of panels (Potential >= target), the request can potentially be met, and the innermost ring 156 is coloured dark grey.

15 Those skilled in the art will appreciate that an alternative and simplified Time-Availability System 1 need not make use of this innermost ring 156.

▪ **Options**

20 The second, middle ring 158, corresponding to the processing of the 'options' bitmap, represents panels on which there exist options. That is, these are panels which have unconfirmed bookings by prospective clients. This allows for a client to provisionally reserve panels for an advertising campaign without the risk and expense of having to commit to purchasing time slots which may subsequently prove to no longer be desirable. Options are typically sold in two week (Monday-Sunday) periods, and if not exercised are set to expire on Friday.

30 Where no contractual commitment has been entered into, the Time-Availability System 1 allows for the allocation of these panels to be overridden by the user. In practice this will only be done to provide panels for proven clients known to purchase advertising time and at a good rate.

35 In the present embodiment, where there is a number of panels in the 'Options' category which exceeds the target value (Options >= target) and therefore the date-time-repetition request can be met by cancelling the associated options that section of the second ring is coloured light blue; contrarily, if the target value exceeds the number of panels in the 'Options' category (Options <= target), the request cannot be met by cancelling options and the section of the second ring is coloured pink.

40 Those skilled in the art will appreciate that an alternative and simplified Time-Availability System 1 need not provide for panel options, and therefore the second ring need not be present. Likewise, the format need not be concentric, not indeed in the form of rings.

▪ **Available**

45 The third, outermost ring 160, corresponding to the processing of the 'available' or 'availability' bitmap, represents how well the target number of panels, as defined in the date-time-repetition request, can be met from panels which are currently available. Four degrees of request-matching are presented, which are colour-coded as follows:

- Good (blue), >=20% above target available
- OK (green), >= target available
- Poor (amber), <= (target -10%) available
- Bad (red), <90% target available, or slot too small

55 An example of the lattermost 'Bad' categorisation is when, for example, the client requests 15 second slots but only 10 second slots remain available.

Active zones are indicated on the 'pie-clock' indicator 152 in a similar fashion to that used in the time-request interface 126. Where the degree to which the target number of panels can be met from available panels varies across an active zone, each active zone is further subdivided by additional (white-coloured) radii. These radii indicate where there is a mismatch or discontinuity in

the extent to which the target number of panels can be met between adjoining sections of the active zone.

5 The pie-clock indicator 152 may be tailored according to the type of user and the corresponding level of access to availability information may be restricted as required. Availability information may therefore be restricted by day-parts. For example, national and local sales teams may be provided with different views and/or ability to use the time-availability system 1 to make resource bookings. This may be required, for example, to prevent a local sales team booking time on prime location display panels ahead of the national sales team working on behalf of a potentially more prestigious client planning a national campaign. Similar restrictions may be made according to time, for example, limiting certain sales teams to using the time-availability system 1 for time periods during certain opening hours. Restrictions may also be made according to location, for example, to limiting sales teams to accessing the time-availability system 1 for resources within their own geographical locality.

15 Those skilled in the art will appreciate that an alternative and simplified Time-Availability System 1 may use only innermost (potential) and outermost (available) rings.

20 Figures 14 shows a series of further detailed examples of the Availability Preview pie-clock of Figure 13. Figure 14(a) shows the pie-clock indicator as per Figure 12, resulting from a request as shown in Figure 9 made to the test data set. The period under consideration is initially the first two weeks of February 2008 i.e. 00:00:00 01/02/2008 – 23:59:59 14/02/2008. In the series of panels that follow, we will see how changes in the availability request affect the availability preview pie-clock.

25 The pie-clock of Figure 14(a) results from a baseline request of 1801 target panels, for a 5 second duration every 1 minute, for a 2 period cycle size. We can see that the inner (potential) ring is dark grey, indicating that the request can (potentially) be met from the total inventory of panels. However, the outer ring shows that although the request can be adequately met for the periods 0-12h and 14-16h on the 24-hour clock face of the 'pie-clock' indicator 152, and clearly met (with a 20% capacity margin) for the period 16-24h, it cannot (immediately) be met for the period 12-14h. However, the middle ring shows that for that time period 12-14h the request may be met by overriding panel reservations which have been taken out as an option.

30 In Figure 14(b) the requested duration has been increased to 15 seconds, all other request parameters remaining unchanged. The period 0-16h is unaffected from Figure 13(a) as described above, however, for the period 16-24h the outermost ring has changed colour from blue ('good') to red ('bad') indicating that the request can no longer be met with currently available panels, either because < 90% of the target can be met or because the available slot size is too small. The red colour of the middle ring indicates that this cannot be overcome by overriding panels reserved as an option.

35 Figure 14(c) shows the effect on the availability preview pie-clock as per Figure 13(b) of reducing the number of target panels to 500. We can see that although the periods 0-12h and 14-16h can now be more easily met (the outermost ring, indicating immediate availability, has changed in colour from green 'OK' to blue 'good'; the middle ring, indicating optional availability, has undergone a colour change from red 'options < target' to blue 'options >= target'), the periods 12-14h and 16-24h are unchanged.

40 In Figure 14(d) the cycle size has been increased to 4 periods. As can be seen, dual pie-clocks are displayed to indicate that there are different extents to which the availability request can be met (or not met) across the calendar time period in question (the exact period to which each pie-clock refers being indicated above the relevant pie-clock). While the left-hand side pie-clock, representing the first two weeks of the period remains unchanged from that for Figure 14(c), there is evidently no longer an issue with meeting the availability request for the second two weeks as the blue colour of the outer ring indicates that the target number of panels can be met for all time periods from panels currently available.

45 In Figure 14(e) the cycle size has been increased to 4 periods. As can be seen, dual pie-clocks are displayed to indicate that there are different extents to which the availability request can be met (or not met) across the calendar time period in question (the exact period to which each pie-clock refers being indicated above the relevant pie-clock). While the left-hand side pie-clock, representing the first two weeks of the period remains unchanged from that for Figure 14(c), there is evidently no longer an issue with meeting the availability request for the second two weeks as the blue colour of the outer ring indicates that the target number of panels can be met for all time periods from panels currently available.

50 Finally, Figure 14(e) shows the result on the availability preview pie-clock of a combined change of reducing the requested duration to 10 seconds while also increasing the number of target

panels to 3100. The left-hand pie-clock, corresponding to the first two weeks of the period in question, we can see that the availability request cannot be fully met. Indeed, for the period 0-16h the match is 'bad'. Evidently, from the colour of the inner ring, the request exceeds the inventory of panels, although for the period 16-24h the match is 'poor' at within 90% of the panels requested – which is equal to the match for all time periods for the second two week period, as shown in the right hand side pie-clock.

Further detail on panel availability can be obtained from the Availability Preview window 150 by either right or left-clicking (mouse motions familiar to those skilled in the art) on parts of the 'pie-clock' indicator 152, thereby displaying tool-tips and pop-up menus.

In the following examples, the pie-clock indicator of availability window 150, as shown in Figures13, is used.

▪ **Tool-tips**

Figure 15 (a) shows the tool-tip 170 displayed by left-clicking on area 153 of the 'pie-clock' indicator 152, providing information identifying the sector under consideration (the two-hour period from 12:00-13:59), the number of available panels (none), and the slot size (60 seconds). The slot size is calculated by subtracting the start seconds from the end seconds and adding 1 second. This is essentially the information stored in the best-match matching period map 80-3.

Figure 15(b) shows the result of right-clicking on area 153 of the 'pie-clock' indicator 152, which presents a mini-menu 172 offering access to further information regarding the corresponding zone 153, ring 160 or information on all rings. Selecting a 'zone detail' menu item summons the zoned detail window 180 described in the flowing section.

▪ **Zone Detail**

Figure 16 shows the zone detail window 180 summoned by right-clicking on the 'pie-clock' indicator 152 and selecting 'All Ring Zones Detail' from the mini-menu 172 as described above. Zone detail window 180 presents all zone data for the corresponding pie-clock indicator in table form. The zone types 'availability' 182, 'option' 184 and 'potential' 186 presented in the table can be seen to correspond to the pie-clock zones 'availability' 160, 'option' 158 and 'potential' 156, respectively. In particular, row 193 can be seen to present the detail of area 153, as shown previously by tool-tip 170. That zone detail window 180 can be seen to present a greater level of detail than is possible with the pie-clock indicator 152 is evident by comparing the information presented in rows 194 and 195 with the corresponding zone 160 as displayed by the pie-clock. Although both rows 194 and 195 relate to the time period in question, and correspond to panels which match the availability query, the pie-clock only shows information corresponding to row 194. This is because the pie-clock presents only the most useful and advantageous information to the user; the panels represented by row 194 are available over a longer time period than those represented by row 195 (from 00:00:00, compared to 00:00:50), are available for a longer time slot size (60 seconds, compared to 10 seconds) and most importantly are greater in number (2066, compared to 1006) so can meet the availability request target, whereas those represented by row 195 cannot.

Further functionality is provided via the menu system, including the ability to 'merge' selected zones (presented as rows), to form composite zones. Also provided (via Applications –Open) is the ability to 'drill down' ring / zone detail to the level of individual panels. This allows to user to access information regarding the panel characteristics, such as the current panel display.

Further alternatives and enhancements

In the present embodiment, all options within the 'options' bitmap are of equal worth. Those skilled in the art will appreciate that in an alternative embodiment options of different worth may be provided, with their value determined according to the desirability of the size and/or timing of the slot and/or to the desirability (or otherwise) of the client.

An enhanced embodiment allows for bidding between clients for slots. For example, client B may choose to reserve an option at a higher price than that offered by client A, whereupon the slot being contested is assigned to client B, and client A is informed that the option has been reallocated and is requested to make a counteroffer.

As has been described above, options represent provisionally reserved resources. Options may be cancelled or overridden if the resources are required to meet a subsequent resource request. In a further embodiment, a distinction is drawn between options taken out by the company operating the time-availability system ('internal' options) and those taken out by clients ('external' options). External options may be taken out by clients directly, for example by means of a web interface, or by employees of the operating company acting on their behalf. In this embodiment, it is possible to cancel or override internal options, but not external options.

In the present embodiment, Availability Preview window 150 presents a 'pie-clock' indicator 152 comprising three concentric rings indicating the 'potential', 'options' and 'available' status. Simpler pie-clock indicators without one or other of the rings have already been discussed. Yet further embodiments may have additional rings for presenting further information regarding resource availability. For example, a four-ring system, wherein the fourth ring is located between the 'option' and 'potential' rings, may be used to provide information regarding bookings.

The innermost 'potential' ring 156, which represents the total inventory of resources, may be shown by the 'pie-clock' indicator 152 only for those time periods of interest as selected by the time-request interface 126. Alternatively, the 'potential' ring 156 may be shown for all time periods.

Furthermore, where the total number of resources varies during the day (for example, as some resources are inactive or inaccessible during certain time periods), the 'potential' ring 156 may itself exhibit detail structure, such as changes in colouration or further subdivision by (white-coloured) radii (as described above with reference to the 'available' ring), indicating where there is a change in the total number of resources. Such radial subdivisions may be used for any ring to indicate a discontinuity in the parameter represented by the ring.

Time slots which remain unassigned may be allocated by default to, for example, public service announcements or for use by charities. Emergency announcements (for example, weather warnings, information on road traffic accidents) may be allowed to override scheduled displays. This may be facilitated by, for example, another dedicated bitmap (as used to introduce the options facility) with priority over all other bitmaps.

The pattern matching facility may be adapted to match for availability by category of content, so as to prevent unwanted clashes of subject matter. For example, rival car manufacturers may want to avoid having their advertising either following one another or within a specified time-frame of each other, with a view to preventing the viewer making product comparisons. Pattern matcher 60 therefore avoids presenting availability matches which breach such pre-defined subject matter exclusion rules.

In an alternative, rather than match to a specific time-availability request, the availability engine 34 may instead be configured to match and display all available slots matching other criteria, such as, for example, all available slots of 10 seconds duration, irrespective of panel characteristics.

The drag-and-drop functionality described previously for selecting a selection template when interacting with the query generation interface 120 to form the time-request can also be extended to allow for user interaction with the pie-clock indicator 152 directly. For example, the user may drag and drop one or more selection templates onto one or more zones of the pie clock indicator 152, and the availability engine will re-compute the availability for the new selection. This allows for interactive sessions where different selection templates (representing different panel or resource groups) can be considered and the effect on the availability immediately assessed.

A further enhancement to the ability to drill-down in the zone detail mode to individual panels allows the user to view the display history of a panel. The historical time/date range of interest may be determined via an interface similar to the query generation interface 120. The display history may be viewed interactively and/or played-back in the manner of a time-lapse sequence. With the addition of a networked webcam, the user can view live images of the panel itself and/or the passing audience.

In the present embodiment, the zone detail window 180 as shown in Figure 14 presents a subset of panel availability information. Further embodiments may present additional information, such as duration and interval information. The information presented may be configurable.

An alternative embodiment which allows for multiple active time zones to be set via the time-request interface 126, envisages pairs of (green) leading edges 142-1 (indicating start time, S) and (red) trailing edges 142-2 of the active zone (indicating end time, E) to be placed on the clock interface by successive mouse clicks i.e. each pair of mouse clicks defining an active zone.

Those skilled in the art will appreciate that while the interaction with the user has been described in terms of the user using a mouse and keyboard, alternatives methods of interaction are possible, for example, via a touch-screen or via voice. Such alternatives are especially useful for facilitating access to the time-availability server 12 and booking system 17 of time-availability system 1 from, for example, mobile handsets, kiosk-style terminals or over the telephone (such as in a telephone-based booking system).

The Java-based web front-end interface as described above, especially in browser-enabled form, is also adaptable for use by clients directly. Naturally, the restricted access and features described in the above may also be used when making access to the time-availability server 12 and booking system 17 of time-availability system 1 available to a wider user base. It is envisioned that a yet more advanced embodiment will allow the client also control the creative part of the process, via access to content provision server 4, and thereby offer an integrated advertisement creation and placement system.

- Offsets

Figures 17 show an enhanced version of the query generation interface 120. In the embodiment described above with reference to Figure 9, the start/stop times of bookings for the display of client advertising content or 'sequences' are constrained by the time-request interface 126 being of a specific granularity (due to the difficulty of showing finer time increments on the time-request interface 126 clearly due to limited screen resolution), for example limited to 15 minute intervals such that sequences can only start/stop at say 09:00; 09:15; 09:30 and so on. As described above, other embodiments may have finer (or coarser) sub-divisions. In an alternative embodiment, and with reference to Figures 17 and 18, a facility is provided by the repetition-request interface 124 (as shown, a selection field 200, although alternatives include, for example, a text box or pull-down menu) to allow the user to indicate a time 'offset' for the start/stop of a client sequence to start/stop at an intermediate time e.g. a 10-minute offset would allow a sequence to start at 09:10.

More specifically, for an initial start time of 09:00:00, Figure 17(a) shows the offset set to zero minutes, which leaves the time period indicator 134 indicating an unchanged start time of 09:00:00; by contrast, Figure 17(b) shows the offset set to 5 minutes, resulting in the time period indicator 134 indicating a start time of 09:05:00.

Figure 18 shows how the availability query 78, comprising parameters $\{N^*, T, R, S, E, D\}$, has parameter S (start time) modified by ΔS (the offset) by the offset function to create a revised query comprising the parameters $\{N^*, T, R, S+\Delta S, E, D\}$.

- Interleaving

As mentioned previously, the pattern matching algorithm determines if the requirement can be met with available panels. This allows for automatic accommodation of multiple client sequence requests which would otherwise only be possible by deliberate manual allocation.

When the 'offset' feature is used in combination with the 'interval' feature this allows client content to be interleaved, providing ability to create long loops of content, i.e. a loop of four minutes would allow multiple 30 second duration content.

For example, if the requirement was to provide for 8 different contents (A-H) each of 30 second duration from 9:00 am to 11:00 am, it could be achieved thus:

	A	B	C	D	E	F	G	H
Start time	9:00	9:00	9:00	9:00	9:00	9:00	9:00	9:00
Offset (mins)	0	0	1	1	2	2	3	3
Effective start time	9:00	9:00	09:01	09:01	09:02	09:02	09:03	09:03
End time	10:59:59	10:59:59	10:59:59	10:59:59	10:59:59	10:59:59	10:59:59	10:59:59
Duration (s)	30	30	30	30	30	30	30	30
Interval (s)	240	240	240	240	240	240	240	240

- wherein content A and B is allocated at the default offset of zero, content C and D is offset by one minute, content E and F by two minutes and content G and H by three minutes.

5 This would result in the following being displayed:

Time	Content							
9:00:00 – 09:00:59	A	B						
9:01:00 – 09:01:59			C	D				
9:02:00 – 09:02:59					E	F		
9:03:00 – 09:03:59							G	H
9:04:00 – 09:04:59	A	B						
9:05:00 – 09:05:59			C	D				
9:06:00 – 09:06:59					E	F		
9:07:00 – 09:07:59							G	H

10 - wherein for the first minute (9:00:00 – 09:00:59) content A and B is shown, for the second minute (9:01:00 – 09:01:59) content C and D is shown, for the third minute (9:02:00 – 09:02:59) content E and F is shown and for the fourth minute (9:03:00 – 09:03:59) content G and H is shown. For the fifth minute (9:05:00 – 09:05:59) the loop returns, showing content A and B and so on for successive minutes. This loop repeats continuously until 10:59:59.

15 It will be understood that the invention has been described above purely by way of example, and modifications of detail can be made within the scope of the invention.

Each feature disclosed in the description, and (where appropriate) the claims and drawings may be provided independently or in any appropriate combination.

20 Reference numerals appearing in the claims are by way of illustration only and shall have no limiting effect on the scope of the claims.

CLAIMS

AVAILABILITY BITMAP

- 5 1. Apparatus for determining the availability of a resource, the apparatus comprising:
means for receiving an input regarding the availability of the resource;
means for determining the availability of said resource; and
means for providing an output regarding the availability of the resource;
10 wherein the means for determining the availability of the resource is adapted to determine the
availability of the resource by means of a resource availability value map.
2. Apparatus according to claim 1, wherein the resource availability value map comprises a
sequence of values.
- 15 3. Apparatus according to claim 2, wherein the position of a value within the sequence represents
a time period.
4. Apparatus according to claim 2 or 3, wherein the value represents the availability of the
resource.
- 20 5. Apparatus according to claim 4, wherein the availability of the resource is represented
numerically.
6. Apparatus according to claim 5, wherein the numerical representation is binary.
- 25 7. Apparatus according to claim 5, wherein the numerical representation is ternary.
8. Apparatus according to claim 5, wherein the numerical representation is of an order greater
than three.
- 30 9. Apparatus according to any of claims 3 to 8, wherein the values are arranged within the
sequence in chronological order.
10. Apparatus according to any of claims 3 to 9, wherein at least one position within the
35 sequence represents a time period in the future.
11. Apparatus according to any of claims 3 to 10, wherein at least one position within the
sequence represents a time period in the past.
- 40 12. Apparatus according to any of claims 3 to 11, further comprising means for periodically
evolving the positions of the values within the sequence.
13. Apparatus according to claim 12, wherein the evolution means is adapted to maintain the
approximate position of a time window, delimited by a first position and a second position in the
45 sequence, with respect to a third position in the sequence.
14. Apparatus according to claim 13, wherein the third position in the sequence represents the
current time period.
- 50 15. Apparatus according to any of the preceding claims, wherein the determining means is
adapted to process a plurality of resource availability value maps in a two-dimensional array.
16. Apparatus according to claim 15, wherein the value maps are arranged in sequence, the
apparatus further comprising means for linking the value map for a particular resource to its
55 position within the sequence.
17. Apparatus according to any of the preceding claims, further comprising means for generating
a resource availability value map from a resource database.

18. Apparatus according to any of the preceding claims, further comprising means for generating resource selection templates for selecting at least one resource.
- 5 19. Apparatus according to claim 18, wherein the resource selection template is generated from resource group information or group membership information.
20. Apparatus according to claim 19, wherein resource group information or group membership information is obtained from a resource template library.
- 10 21. Apparatus according to any of claims 19 or 20, wherein resource group information or group membership information comprises information regarding physical characteristics or attributes of the resource.
- 15 22. Apparatus according to any of claims 19 to 21, wherein resource group information or group membership information comprises information regarding administrative characteristics or attributes of the resource.
- 20 23. Apparatus according to any of claims 18 to 22, wherein the resource selection templates comprise a sequence of values.
- 25 24. Apparatus according to any of claims 18 to 23, further comprising means for identifying the at least one resource selected by the resource selection template.
26. Apparatus according to claim 24, wherein the resource is identified by the position of the value in the sequence.
- 30 27. Apparatus according to any of claims 18 to 26, further comprising means for combining selection templates.
28. Apparatus according to any of claims 15 to 27, further comprising a resource availability value map library, for storing group-specific resource availability value map arrays.
- 35 29. Apparatus according to claim 28, wherein the resource availability value map array is stored as a sequence of record elements.
- 40 30. Apparatus according to claim 29, wherein each record element represents the availability of the resources during a time period.
- 45 31. Apparatus according to claim 30, wherein each record element comprises a start period field, representing the start of the time period.
32. Apparatus according to claim 30 or 31, wherein each record element comprises an end period field, representing the end of the time period.
33. Apparatus according to any of claims 30 to 32, wherein each record element comprises a value map field, representing the availability of the resources during the time period.
- 50 34. Apparatus according to any of claims 30 to 33, wherein each record element comprises a period lock field.
- 55 35. Apparatus according to claim 34, wherein the period lock field is a mutual-exclusion or mutex lock.
36. Apparatus according to any of the preceding claims, wherein the input regarding the availability of the resource is in the form of a query as to whether there are resources available to fulfil a resource requirement specified in the query.

37. Apparatus according to any of claims 1 to 36, wherein the input regarding the availability of the resource is in the form of a generic query to determine what resources are available, without requesting specifying a resource requirement
- 5 38. Apparatus according to any of the preceding claims, wherein the input means further comprises means for accepting a resource booking request.
39. Apparatus according to claim 38, wherein the means for accepting the resource booking request is operable to set at least one value in the resource availability value map for the
10 corresponding resource, for the corresponding time period specified in the booking request.
40. Apparatus according to claim 39, wherein the value is set subject to the resource being available.
- 15 41. Apparatus according to any of the preceding claims, further comprising means for representing booked resources in a value map.
42. Apparatus according to any of the preceding claims, further comprising means for representing resources for which exist options in a value map.
- 20 43. Apparatus according to claim 42, wherein the value map represents the availability of a resource which has been booked with a predetermined degree of certainty.
- 25 44. Apparatus according to claim 42 or 43, wherein all options within the option value map have equal value.
45. Apparatus according to claim 42 or 43, wherein options within the option value map have a plurality of values.
- 30 46. Apparatus according to claim 45, comprising a plurality of option value maps, each option value map representing resources which have been booked with a degree of certainty specific to that option map.
- 35 47. Apparatus according to any of claims 43 to 46, wherein the degree of certainty of a booking depends on a characteristic of the entity making the booking.
48. Apparatus according to any of claims 43 to 47, wherein the degree of certainty of a booking depends on a characteristic of the resource.
- 40 49. Apparatus according to any of claims 43 to 48, wherein the degree of certainty of a booking depends on the time period booked.
- 45 50. Apparatus according to any of claims 43 to 49, wherein the booking certainty is represented by a value.
51. Apparatus according to any of claims 43 to 50, wherein the availability query comprises an option override value.
- 50 52. Apparatus according to any of claims 50 to 51, wherein the determining means is adapted to determine whether or not a provisional booking may be overridden by comparison of the option override value with the booking certainty value.
- 55 53. Apparatus according to any of the preceding claims, further comprising data synchronising means for synchronising the availability data between the resource availability value maps and the resource database.
- MEMORY SCRUBBING
- 60 54. Apparatus for processing a resource availability value map array, the apparatus comprising:
means for comparing at least two record elements of the value map;

means for determining whether said record elements can be validly combined; and
means for combining said record elements if they can be validly combined.

- 5 55. Apparatus according to claim 54, wherein the determining means is adapted to determine that the record elements can be validly combined if the record elements refer to adjacent time periods.
- 10 56. Apparatus according to any of claims 54 to 55, wherein the determining means is adapted to determine that the record elements can be validly combined if the record elements indicate identical resource availabilities.
- 15 57. Apparatus according to any of claims 54 to 56, wherein the determining means is adapted to process the resource availability value map array repeatedly.
- 20 58. Apparatus according to any of claims 54 to 57, wherein the determining means is adapted to process the resource availability value map according to predetermined conditions.
- 25 59. Apparatus according to any of claims 54 to 58, wherein the determining means is adapted to process the resource availability value map according to a schedule.
- 30 60. Apparatus according to any of claims 54 to 58, wherein the determining means is adapted to process the resource availability value map according to an internal triggering event.
- 35 61. Apparatus according to claim 60, wherein the internal triggering event is a change in the value map array detected by the apparatus.
- 40 62. Apparatus according to any of claims 54 to 58, wherein the determining means is adapted to process the resource availability value map according to an external triggering event.
- 45 63. Apparatus according to claim 62, wherein the external triggering event is in the form of a command issued by the user or by some further apparatus.
- 50 64. Apparatus according to any of claims 54 to 63, wherein the means for combining said record elements combines first and second record elements by:
i) setting the value of the end period field of the first record element to the value of the end period field of the second record element; and
ii) disposing of the second record element.
- 55 65. Apparatus according to claim 64, wherein the record elements are compared and combined two at a time.
66. Apparatus according to claim 64, wherein a plurality of record elements are compared at the same time.
67. Apparatus according to claim 66, wherein the plurality of record elements are combined at the same time.
68. Apparatus according to claim 66, wherein the plurality of record elements are combined sequentially by means of an iterative process.
- 50 CONSTRUCTION OF RESOURCE AVAILABILITY QUERY
- 55 69. Apparatus for constructing a resource time-availability query, the apparatus comprising:
means for generating a user interface comprising a time-based representation;
means for determining, from the interaction of the user with the interface, the required parameters for the resource time-availability query; and
means for constructing a resource time-availability query from said determined parameters; wherein
the required parameters comprise resource availability start and end times.

70. Apparatus according to claim 69, wherein the generating means includes means for permitting the user to adjust the availability start and end times on the time-based representation.

71. Apparatus according to claims 69 to 70, wherein the constructing means constructs a time-availability query further comprising at least some of the following parameters:

- target number of resources
- duration of required availability
- availability repetition interval
- availability date

72. Apparatus according to any of claims 69 to 70, wherein the time-based representation interface comprises one or more of:

- a date indicator
- a time-of-day indicator
- offset selector

73. Apparatus according to any of claims 69 to 72, wherein the determining means is operable to determine, from the user interaction with the date indicator, the dates for which availability is to be queried.

74. Apparatus according to any of claims 72 to 73, wherein the time-of-day indicator is divided into zones, each zone demarcating a time period of availability.

75. Apparatus according to claim 74, wherein the time-of-day indicator comprises first and second user-adjustable markers, and the determining means determines the start and end times from the position of the first and second markers respectively.

PATTERN MATCHING

76. Apparatus for processing a resource availability value map in dependence on a resource time-availability query, the apparatus comprising:

- means for receiving a resource availability value map;
- means for receiving a resource time-availability query; and
- means for matching the requirements of the resource time-availability query to the resource availability represented by the resource availability value map.

77. Apparatus according to claim 76, wherein the matching means further comprises means for outputting the result of at least one match to at least one matching period value map.

78. Apparatus according to any of claims 76 to 77, wherein the matching period value map comprises at least one of the following parameters:

- a map start time period
- a map end time period
- a slot size
- a map status
- a map object type
- the last merged start period for map
- the resource availability value map

79. Apparatus according to any of claims 76 to 78, wherein the outputting means builds the matching period value maps iteratively.

80. Apparatus according to claim 79, wherein the iteration is between start and end time periods.

81. Apparatus according to claim 80, wherein for each time period, the matching means compares the resources specified in the resource time-availability query with the resource availability represented by the resource availability value map for said time period.

82. Apparatus according to any of claims 77 to 81, wherein the apparatus further comprises means for storing the matching period value maps in a matching period map store.

83. Apparatus according to any of claims 76 to 82, wherein the apparatus is adapted to match a matching period value map to a subset of pre-existing records in the form of matching period value maps, selected according to partial matching criteria.

5

84. Apparatus for matching a record to members of an array of pre-existing records according to matching criteria, comprising:

means for selecting a subset of the pre-existing records according to a first part of the matching criteria; and

10

means for comparing the said record to the selected subset of the pre-existing records according to a second part of the matching criteria.

85. Apparatus according to claim 83 or 84 wherein the said record is a matching period value map

15

86. Apparatus according to claim 83, 84 or 85 wherein the pre-existing records are previously matched matching period value maps.

87. Apparatus according to any of claims 83 to 86 wherein the matching criteria are temporal criteria.

20

88. Apparatus according to any of claims 83 to 87 wherein the first and/or second parts of the criteria comprise any of the following: year, season, month, date, day, hour, minute, second.

25

89. Apparatus according to any of claims 83 to 88 further comprising means for generating a parameter associated with each pre-existing record according to the first part of the matching criteria.

90. The apparatus of claim 89 further comprising means for generating a table from the pre-existing records, indexed according to the generated parameter.

30

FILTERING

91. Apparatus for processing comparing matching period value maps, the apparatus comprising:
 means for receiving a first period value map and a second period value map;
 means for comparing a first period value map and second period value map; and
 means for reallocating resources between first period value map and second period value map in dependence on at least one pre-determined requirement.

35

92. Apparatus according to claim 91, wherein the requirement comprises one or more of the following:

- that resources are available for the longest possible consecutive period of time
- that periods of resource availability are preferably contiguous in time and not fragmented
- that the number of available resources during a time period is constant

45

93. Apparatus according to claim 92, wherein the list of requirements comprises one or more further sets of requirements which are used singly or in combination, either with other members of a set or with the original set of requirements.

50

OUTPUT OF RESOURCE AVAILABILITY RESULTS

94. Apparatus for determining the extent to which the availability of a resource matches a resource availability request, the apparatus comprising:

means for comparing the resource availability request with the availability of the resource;

55

and

means for outputting a signal representing the extent to which the availability of the resource matches the resource availability request.

95. Apparatus according to claim 94, wherein the representation signal represents the resource availability as a function of time.

60

- 5 96. Apparatus according to claim 94 or 95, wherein the outputting means further comprises display means for generating a time-based representation in dependence on the representation signal.
97. Apparatus according to any of claims 94 to 96, wherein the outputting means is operable to generate a plurality of representation signals regarding the degree to which the resource availability request can be fulfilled by the available resources at a plurality of times.
- 10 98. Apparatus according to claims 96 or 97, wherein the display means is operable to generate a time-based representation in dependence on the plurality of representation signals and each representation signal is represented in an area of the time-based representation.
- 15 99. Apparatus according to claim 98, wherein the area is further divided into zones according to the variation of resource availability during the time period specified in the resource availability request.
- 20 100. Apparatus according to claim 99, wherein each zone indicates a different resource availability.
- 25 101. Apparatus according to any of claims 99 to 100, wherein the outputting means allocates to each zone of the area an attribute according to whether the resource availability request can be:
- met, with 50%, 20%, 10%, 5%, or 1% spare resource capacity; or
 - met, with less than 20%, 10%, 5%, or 1% spare resource capacity; or
 - not met, but within 10%, 5%, or 1% of the requested resources; or
 - not met, and by more than 1%, 5%, 10%, 20%, or 50% of the requested resources
- 30 102. Apparatus according to any of claims 94 to 101, wherein the outputting means is further operable to generate data regarding the degree to which the resource availability request could be fulfilled by resources on which there exists an option.
- 35 103. Apparatus according to any of claims 98 to 102, wherein the outputting means outputs a signal representing this data, and the display means generates a time-based representation in dependence on this signal in a further area of the time-based representation.
- 40 104. Apparatus according to claim 103, wherein the further area is further divided into zones according to the variation of resource options during the time period specified in the resource availability request.
- 45 105. Apparatus according to claim 104, wherein the outputting means allocates to each zone of the further area an attribute according to whether or not the resource availability request can be met by overriding options.
- 50 106. Apparatus according to claim 105, wherein the outputting means is further operable to generate data regarding the degree to which the resource availability request could be fulfilled if the total number of resources were available and to output a signal representing this data.
- 55 107. Apparatus according to claim 106, wherein the display means generates a time-based representation in dependence on the signal in a yet further area of the time-based representation.
108. Apparatus according to claim 107, wherein the outputting means generates only one area of the time-based representation.
109. Apparatus according to claim 103, wherein only two areas of the time-based representation are generated.
- 60 110. Apparatus according to any of claims 98 to 109, wherein the areas are circular.
111. Apparatus according to any of claims 98 to 110, wherein the zones are segments of a circle.

112. Apparatus according to any of claims 98 to 111, wherein the circles are concentric.

113. Apparatus according to any of claims 98 to 112, wherein the time-based representation is in the form of a clock.

5

114. Apparatus according to claim 113, wherein the clock is marked into twelve –hour divisions

115. Apparatus according to claim 114, wherein more the clock is marked into twenty-four hour divisions.

10

116. Apparatus according to any of claims 94 to 115, wherein the apparatus further comprises means for processing a plurality of period value maps and selecting the period value map which best matches the resource availability request.

15

117. Apparatus according to claim 116, wherein the outputting means is operable to generate data relating to the period value maps processed and to output a signal representing this data.

DRAG & DROP FEATURE

20

118. Apparatus for modifying a time-availability query, the apparatus comprising:
means for receiving a first time-availability query;
means for receiving a resource selection template; and
means for generating a second time-availability query by applying the resource selection template to the first time-availability query.

25

119. Apparatus according to claim 118, further comprising means for further processing the query.

30

120. Apparatus according to claim 119, wherein further processing comprises evaluating the resource availability in accordance with the query.

121. Apparatus according to claim 120, wherein the query may be further modified by applying further resource selection templates, each successive modification being evaluated in turn.

35

METHODS

122. A method of determining the availability of a resource, the method comprising:
receiving an input regarding the availability of the resource;
determining the availability of said resource; and
providing an output regarding the availability of the resource;
wherein the method of determining the availability of the resource comprises determining the availability of the resource by means of a resource availability value map.

40

123. A method of processing a resource availability value map array, the method comprising:
comparing at least two record elements of the value map;
determining whether said record elements can be validly combined; and
combining said record elements if they can be validly combined.

45

124. A method of constructing a resource time-availability query, the method comprising:
generating a user interface comprising a time-based representation;
determining, from the interaction of the user with the interface, the required parameters for the resource time-availability query; and
constructing a resource time-availability query from said determined parameters; wherein the required parameters comprise resource availability start and end times.

50

55

125. A method of processing a resource availability value map in dependence on a resource time-availability query, the method comprising:
receiving a resource availability value map;
receiving a resource time-availability query; and

matching the requirements of the resource time-availability query to the resource availability represented by the resource availability value map.

- 5 126. A method of matching a record to an array of pre-existing records according to matching criteria, comprising:
selecting a subset of the pre-existing records according to a first part of the matching criteria; and
10 comparing the record to the selected subset of the pre-existing records according to a second part of the matching criteria.
127. A method of processing matching period value maps, the method comprising:
receiving a first period value map and a second period value map;
15 comparing a first period value map and second period value map; and
reallocating resources between the first period value map and the second period value map in dependence on at least one pre-determined requirement.
- 20 128. A method of determining the extent to which the availability of a resource matches a resource availability request, the method comprising:
comparing the resource availability request with the availability of the resource; and
outputting a signal representing the extent to which the availability of the resource matches the resource availability request.
- 25 129. A method of modifying a time-availability query, the method comprising:
receiving a first time-availability query;
receiving a resource selection template; and
generating a second time-availability query by applying the resource selection template to the first time-availability query.
- 30 130. A computer program product comprising software code adapted, when executed on a data processing apparatus, to perform a method as claimed in any of claims 122 to 129.
131. A system adapted to perform a method as claimed in any of claims 122 to 129.
- 35 132. A system substantially as described herein with reference to and/or as shown in any of the accompanying drawings.
133. Apparatus substantially as described herein with reference to and/or as shown in any of the accompanying drawings.
- 40 134. A method substantially as described herein with reference to and/or as shown in any of the accompanying drawings.

1 / 17

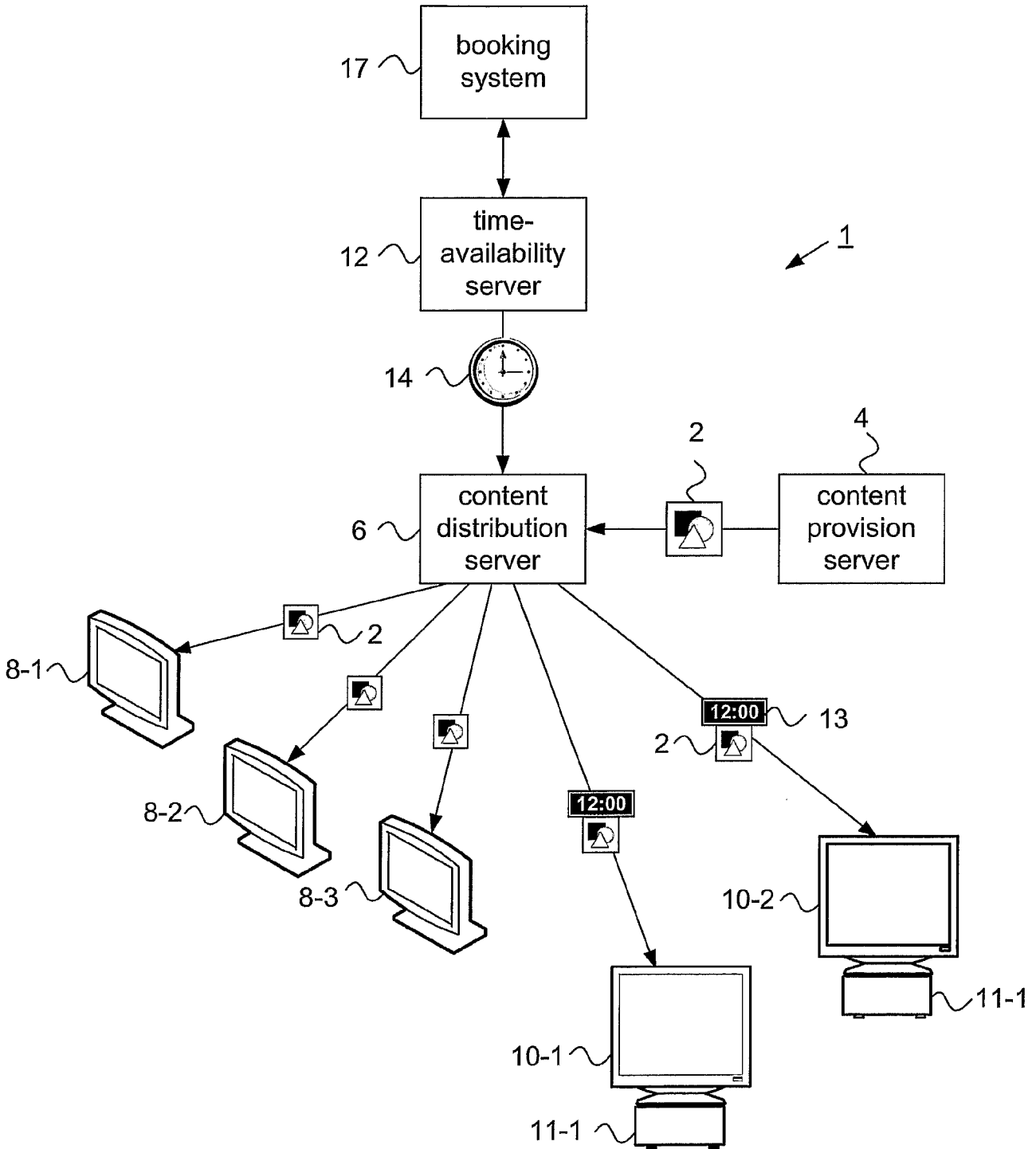
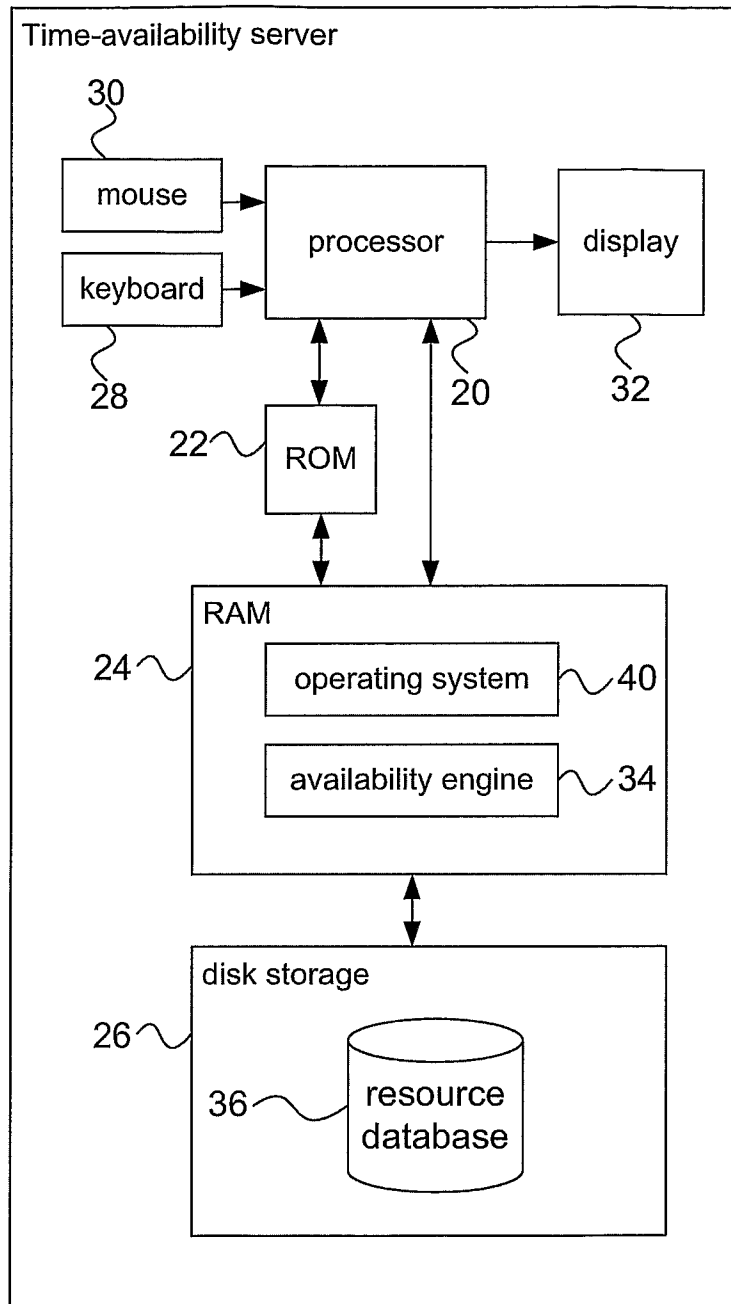


Fig. 1



12

Fig. 2

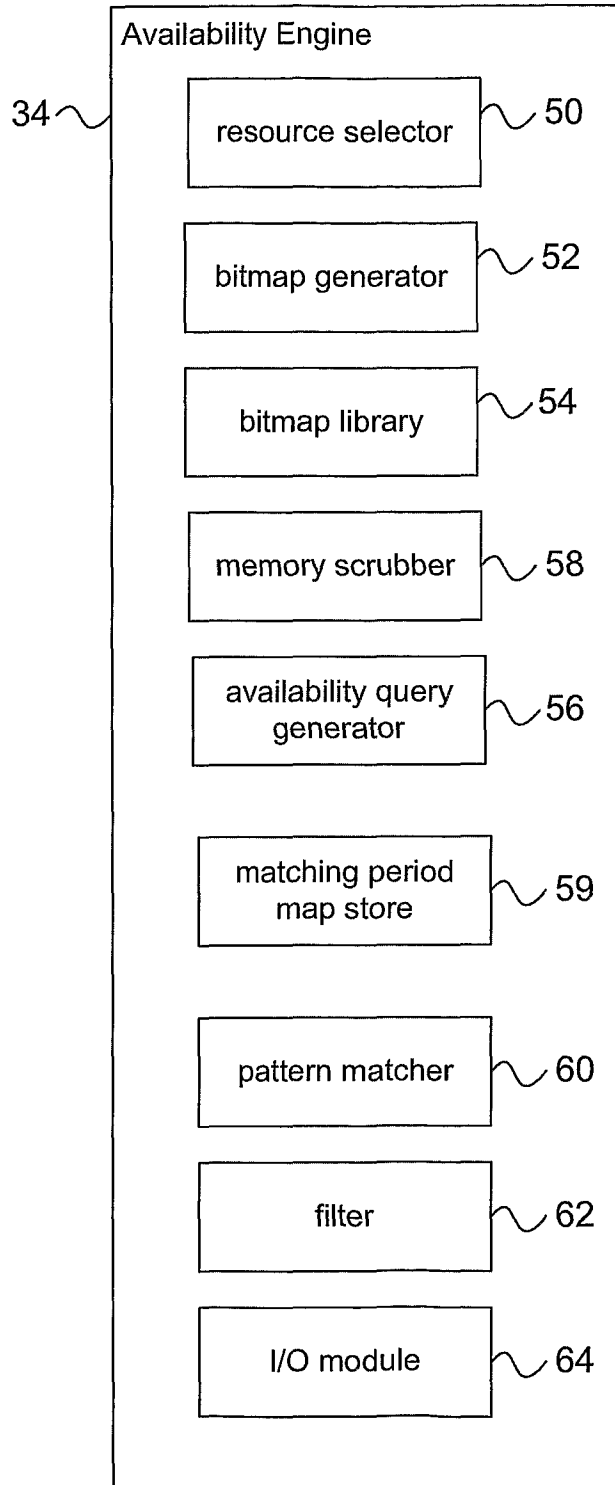


Fig. 3

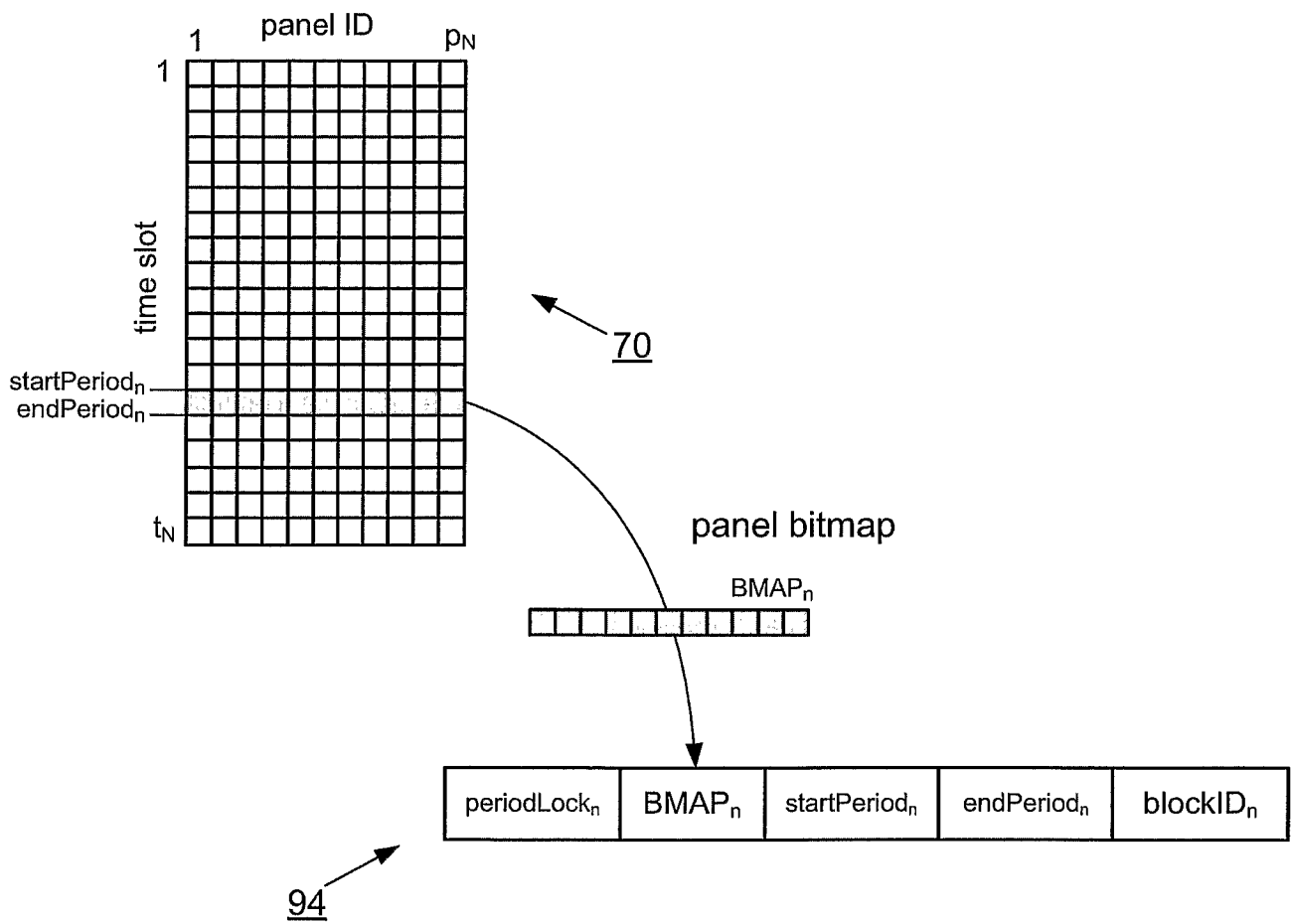


Fig. 5

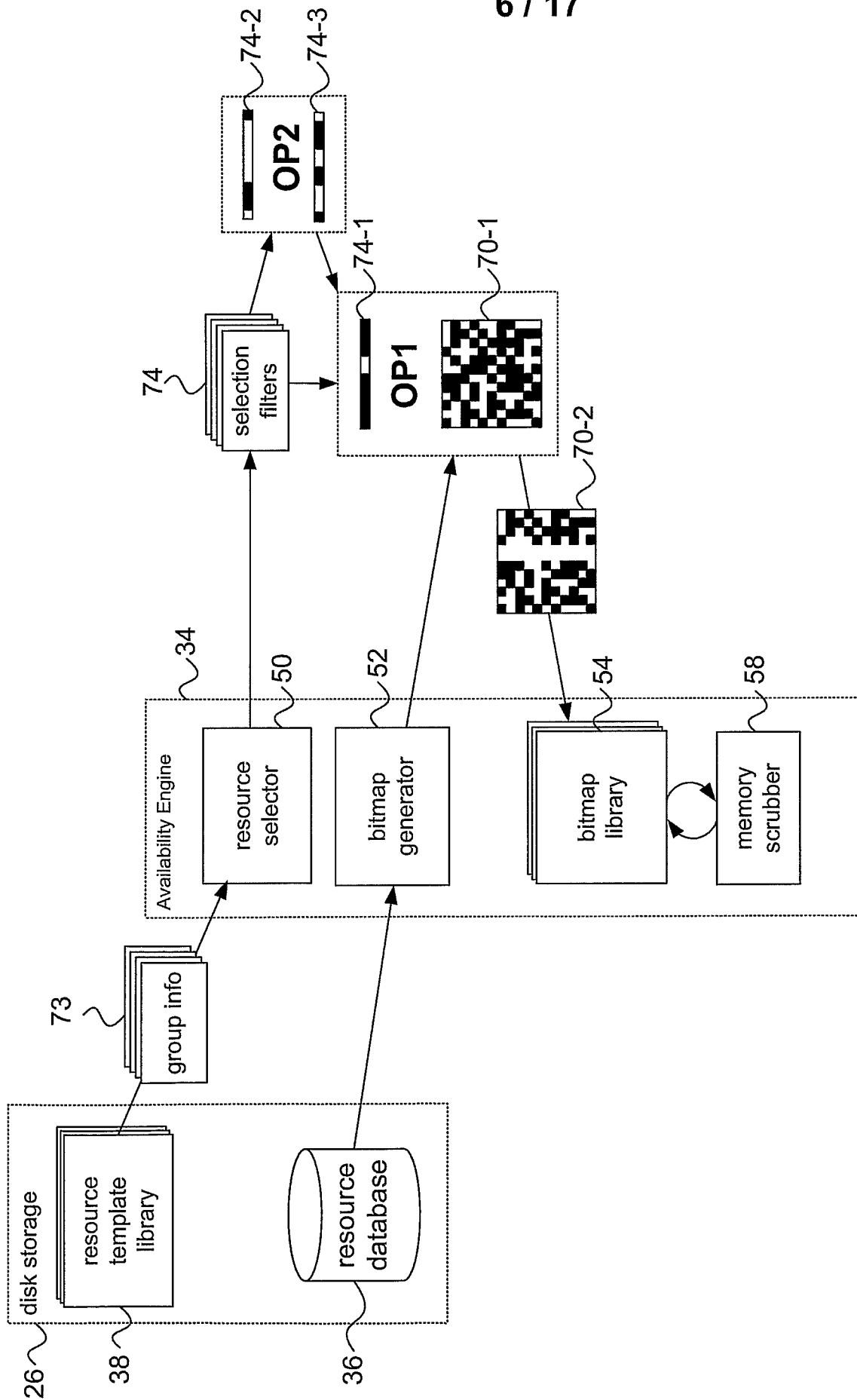


Fig. 6

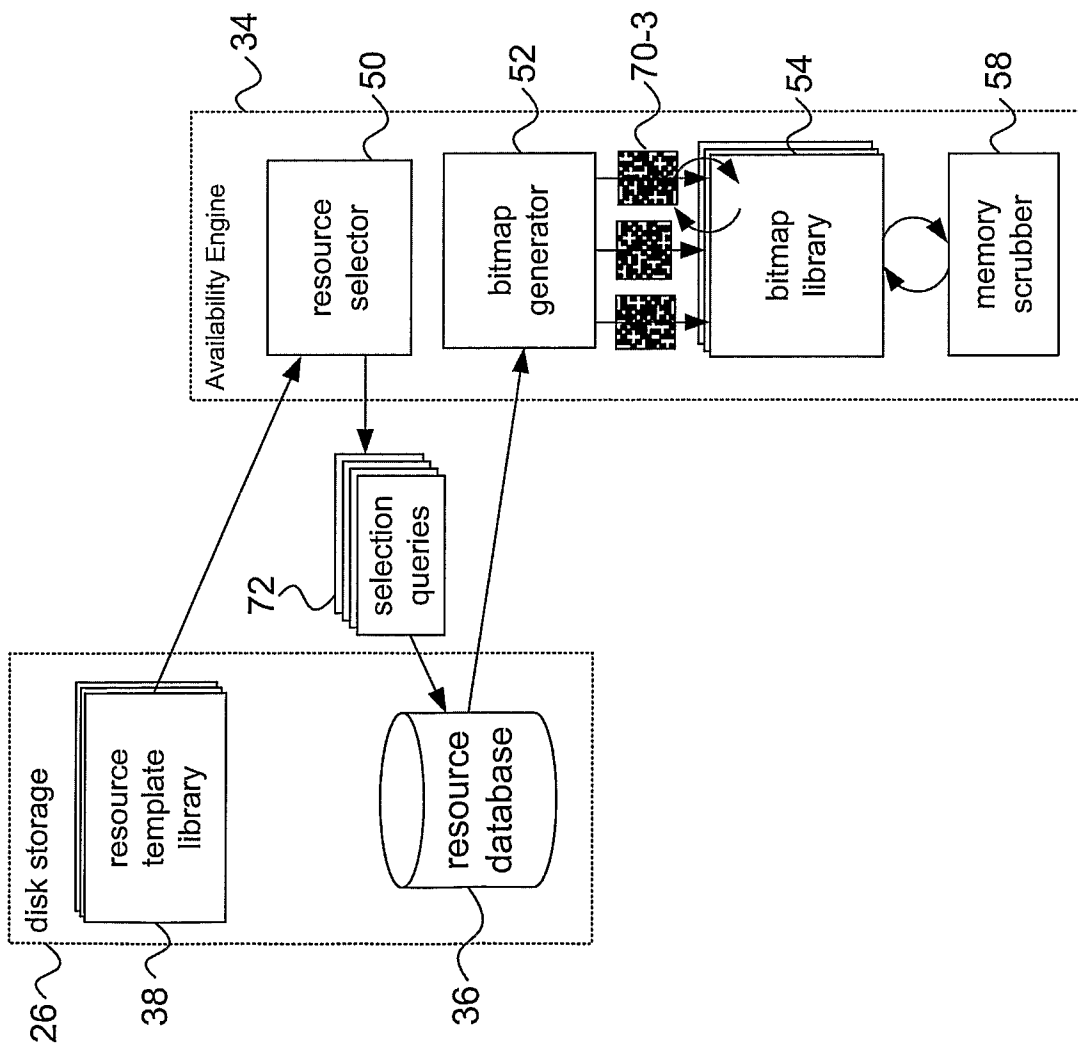


Fig. 7

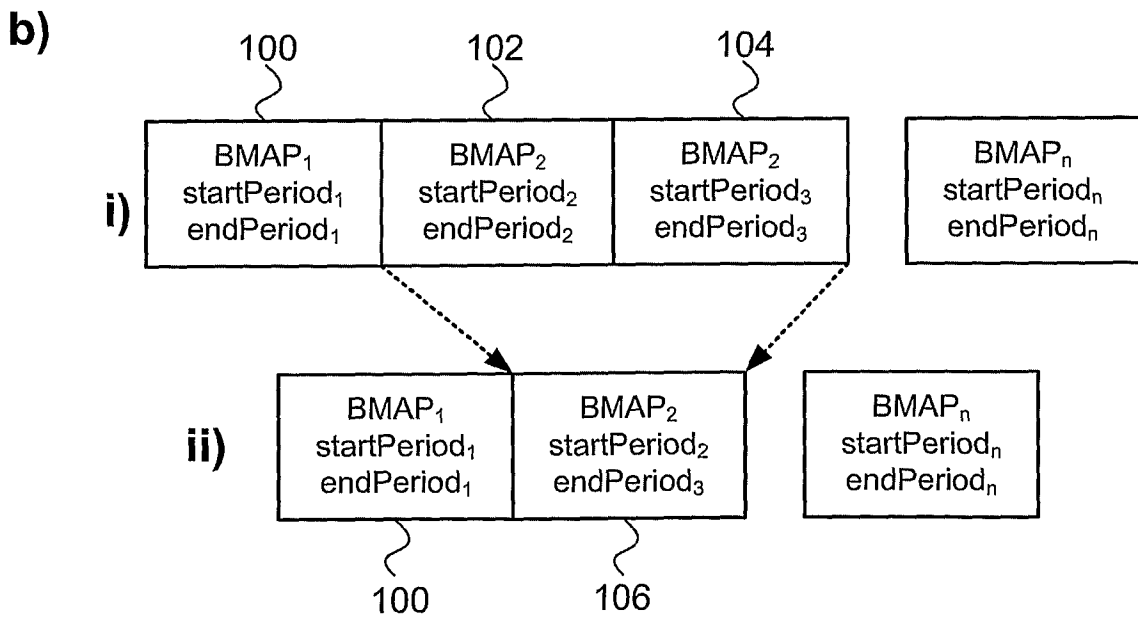
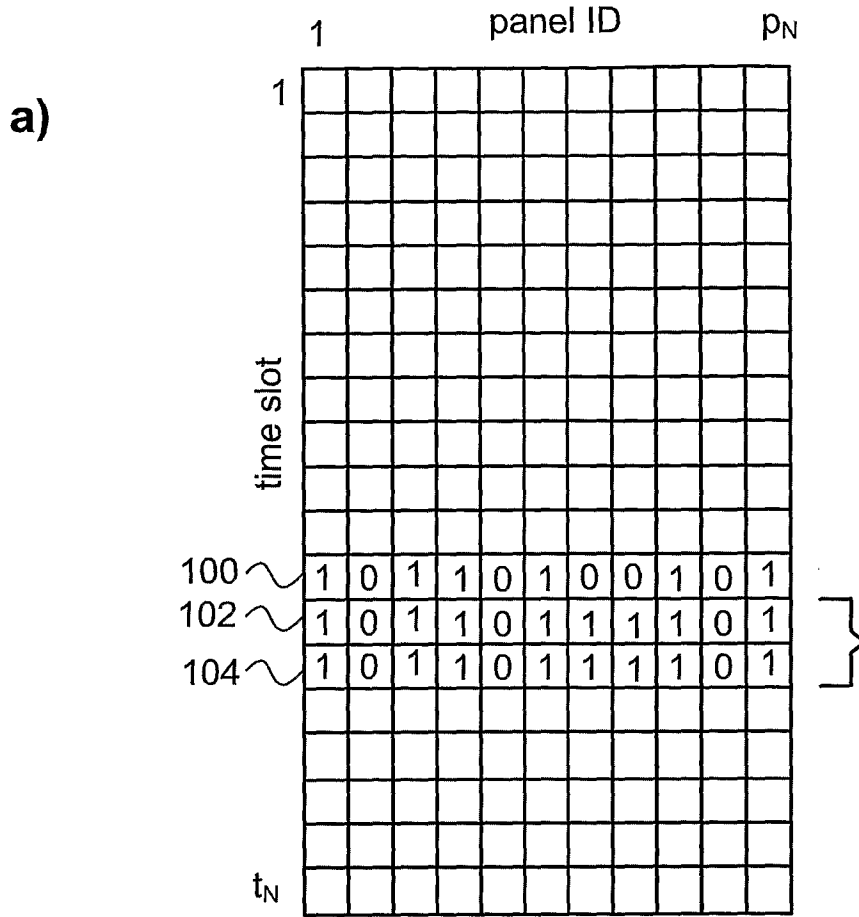


Fig. 8

9 / 17

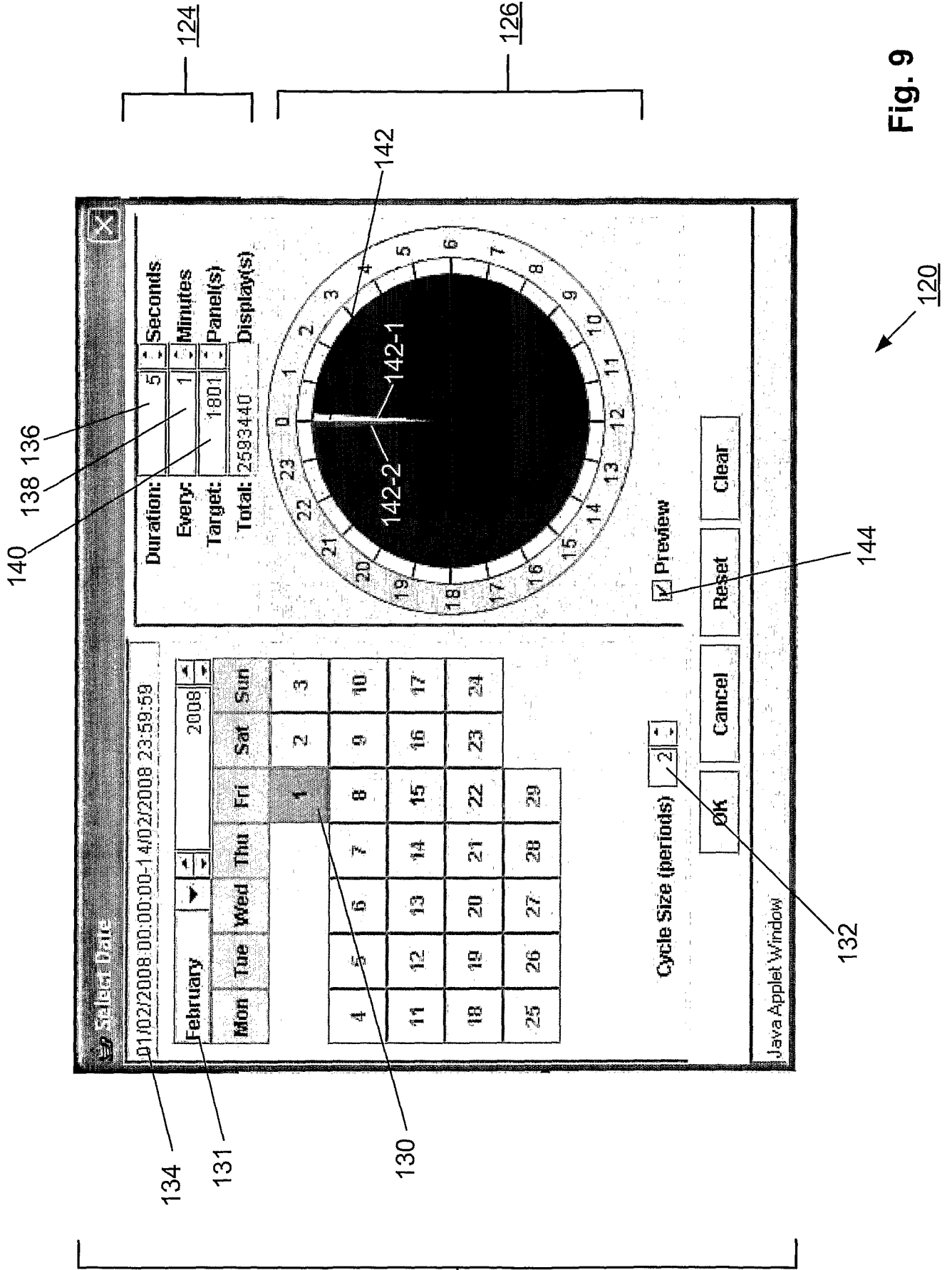


Fig. 9

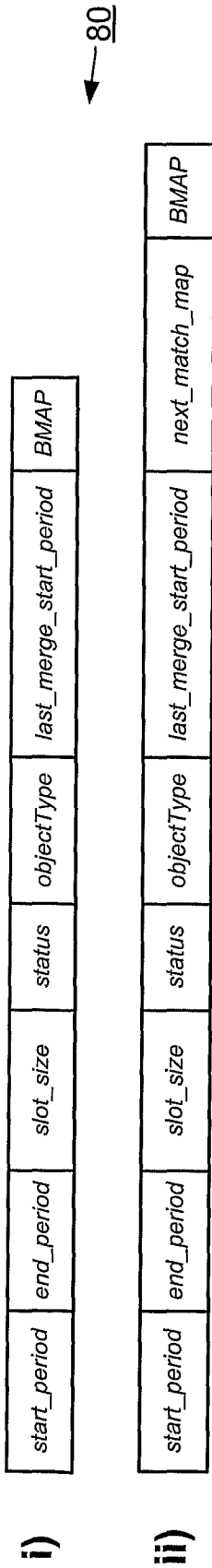


Fig. 11

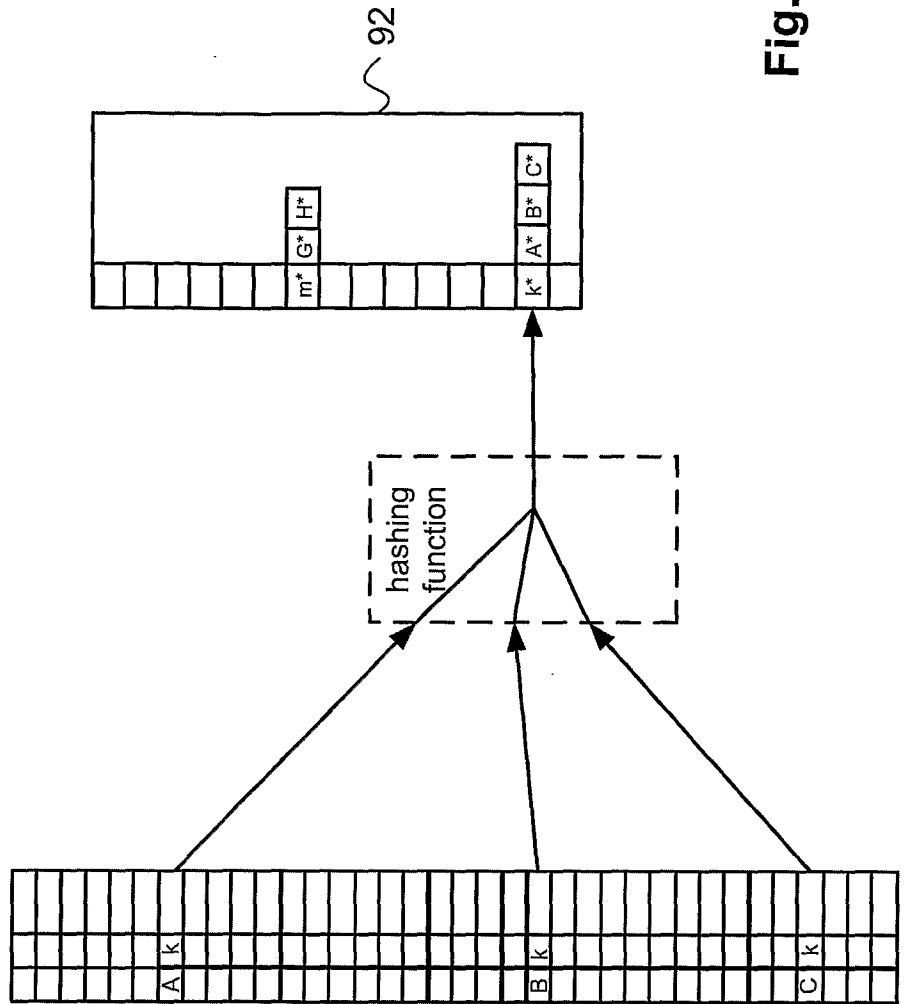


Fig. 12

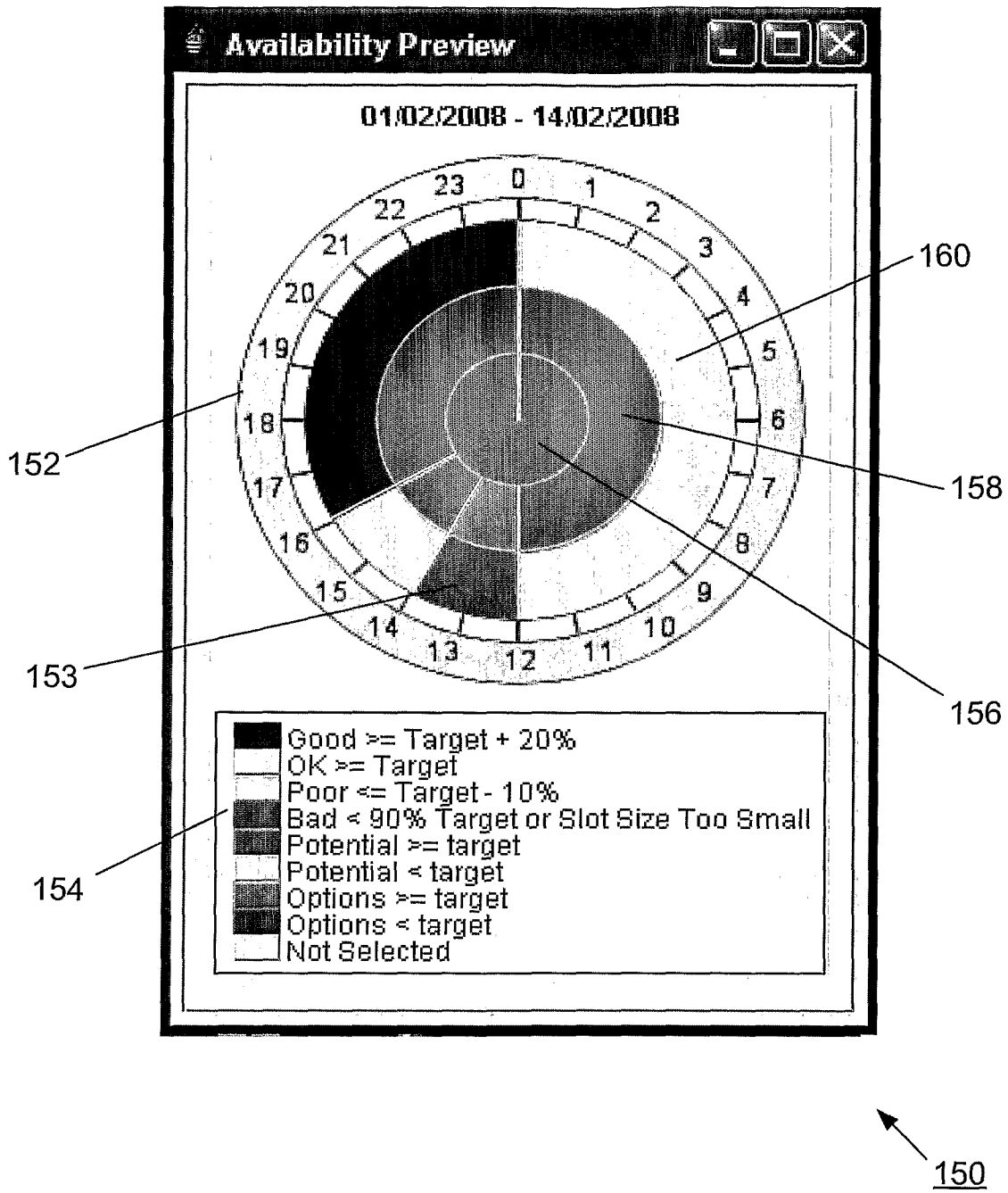
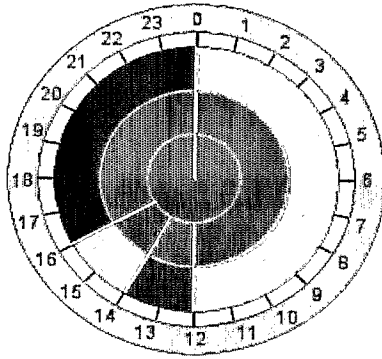
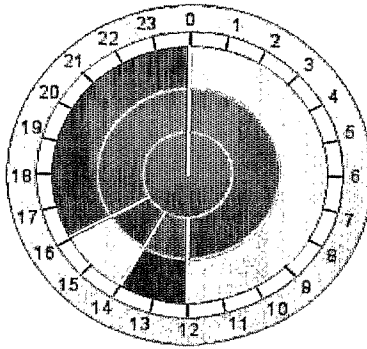


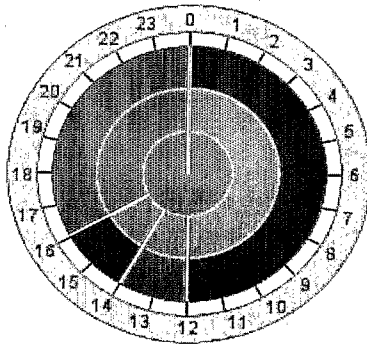
Fig. 13



a) Duration: 5 sec
 Every: 1 min
 Target: 1801 panels
 Cycle size: 2 periods

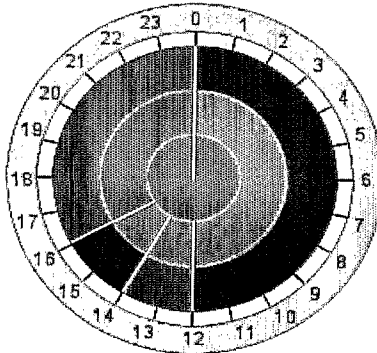


b) Duration: 15 sec
 Every: 1 min
 Target: 1801 panels
 Cycle size: 2 periods

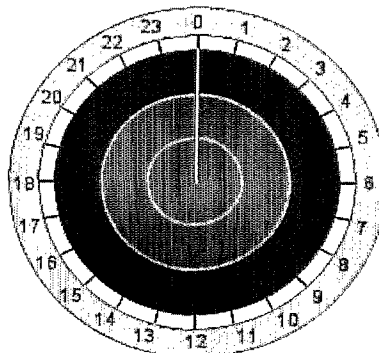


c) Duration: 15 sec
 Every: 1 min
 Target: 500 panels
 Cycle size: 2 periods

01/02/2008 - 14/02/2008

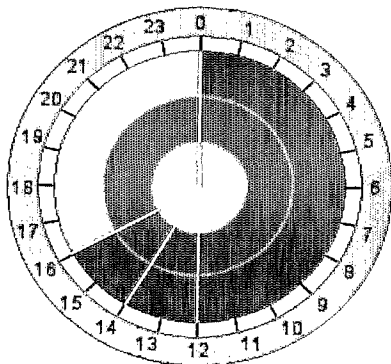


15/02/2008 - 29/02/2008

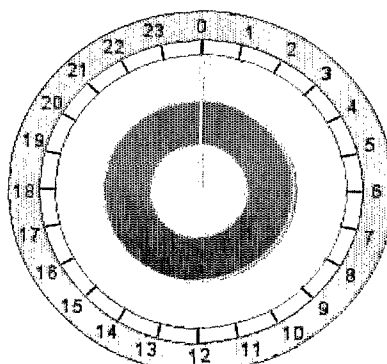


d) Duration: 15 sec
 Every: 1 min
 Target: 500 panels
 Cycle size: 4 periods

01/02/2008 - 14/02/2008



15/02/2008 - 29/02/2008



e) Duration: 10 sec
 Every: 1 min
 Target: 3100 panels
 Cycle size: 4 periods

Fig. 14

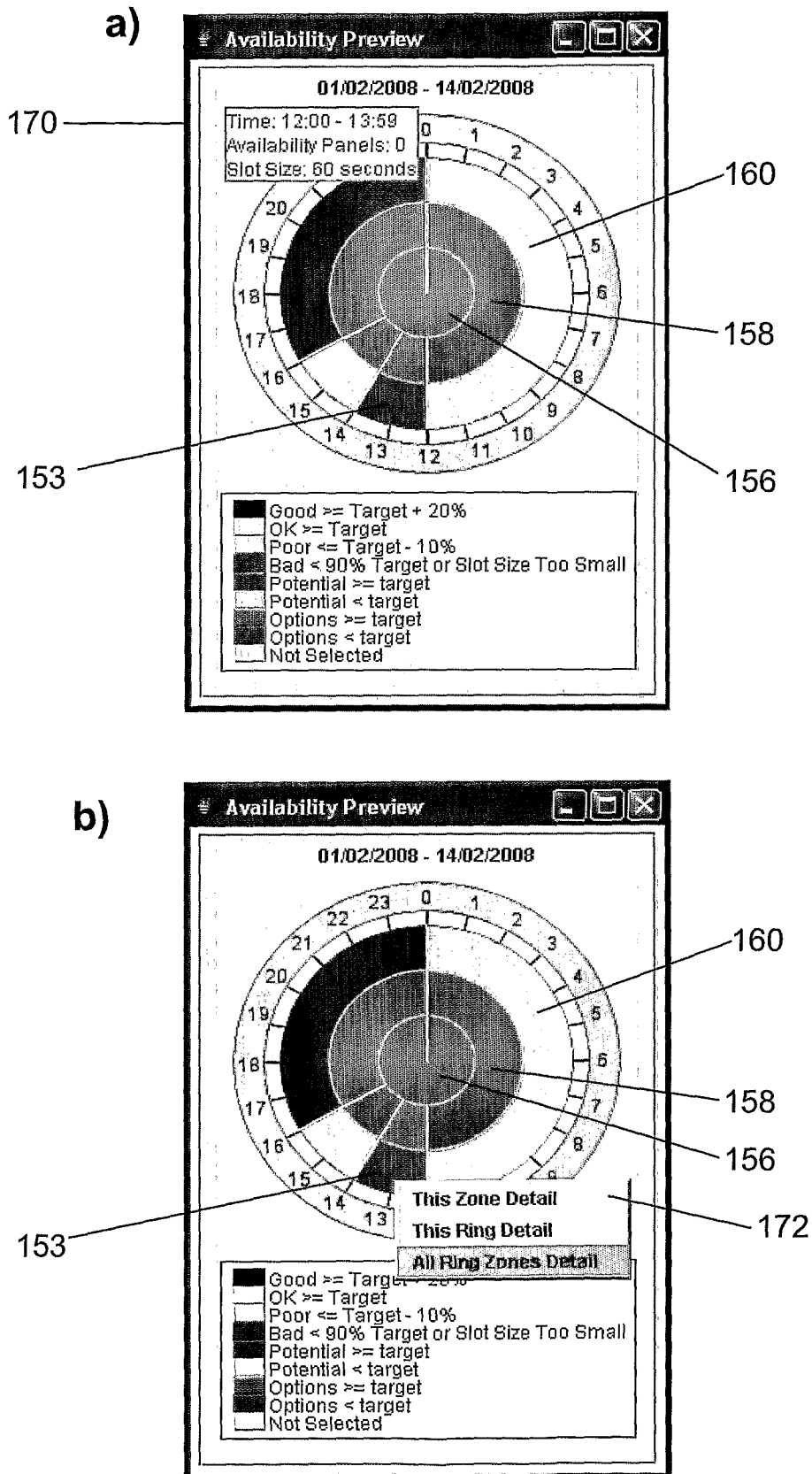


Fig. 15

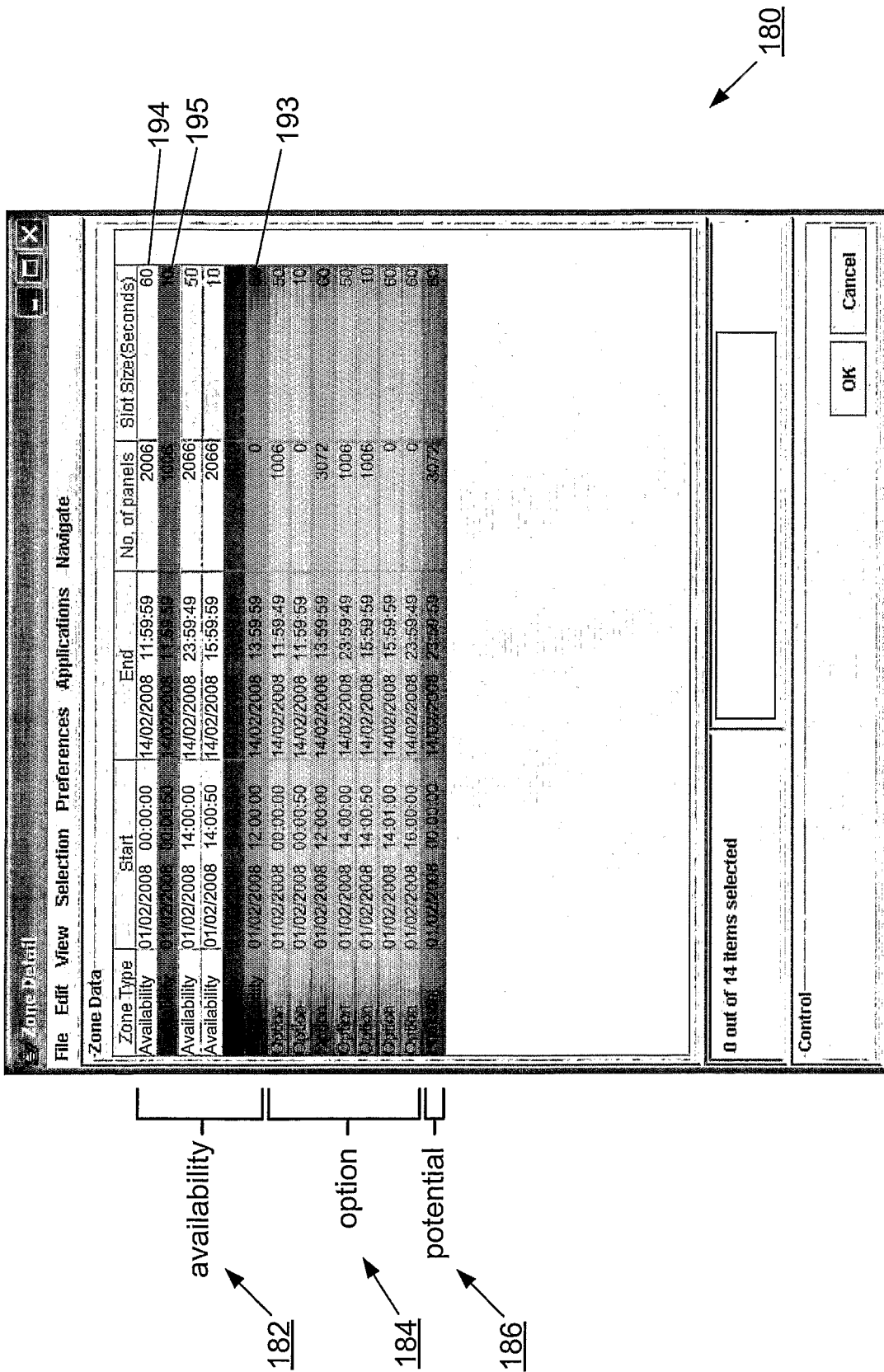


Fig. 16

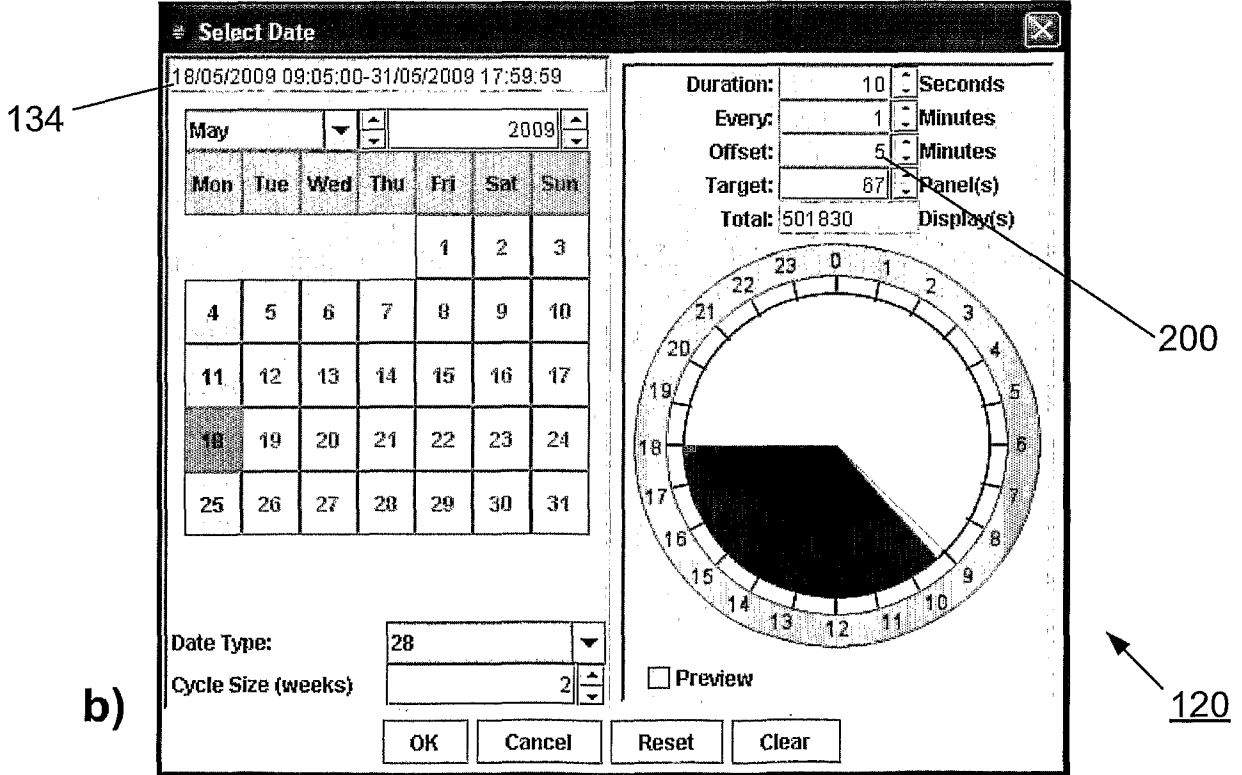
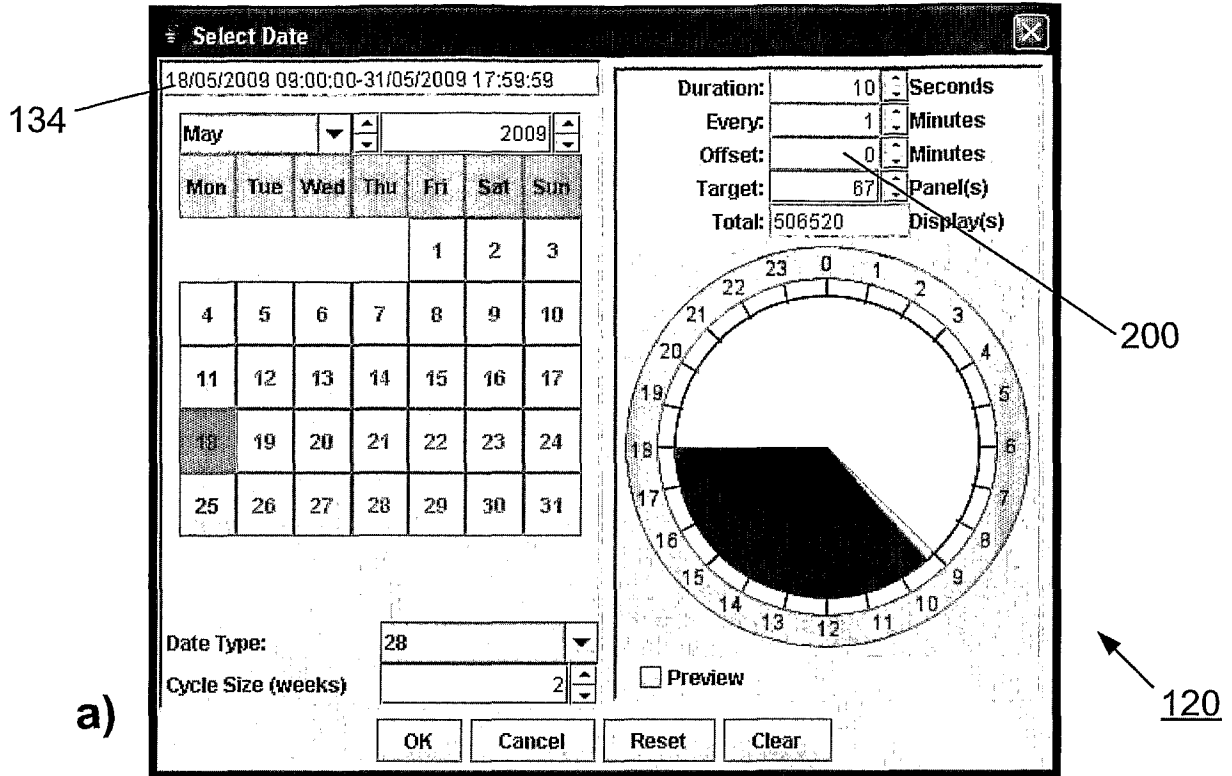


Fig. 17

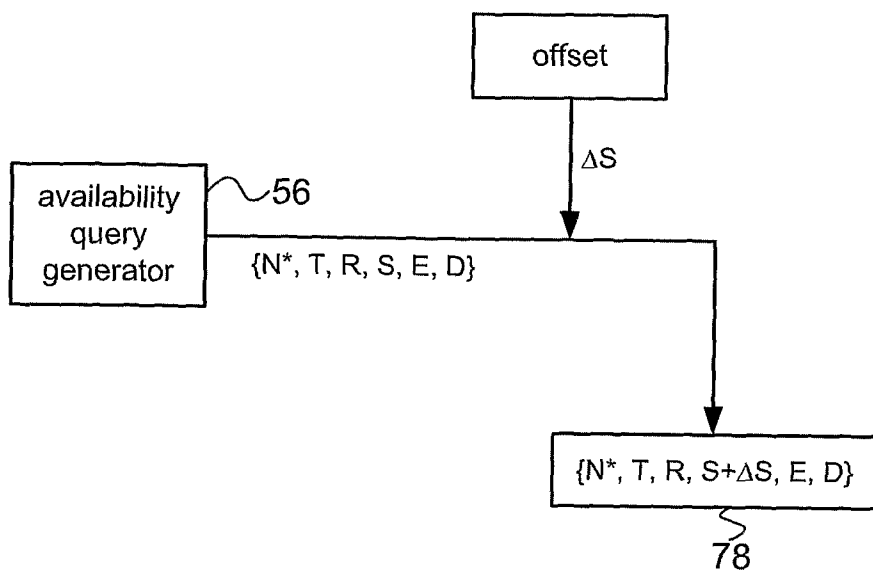


Fig. 18

PATENT COOPERATION TREATY

PCT

DECLARATION OF NON-ESTABLISHMENT OF INTERNATIONAL SEARCH REPORT

(PCT Article 17(2)(a), Rules 13ter.1(c) and Rule 39)

Applicant's or agent's file reference P32034WO/PDC	IMPORTANT DECLARATION	Date of mailing(day/month/year) 14/10/2009
International application No. PCT/GB2009/001333	International filing date(day/month/year) 28/05/2009	(Earliest) Priority date(day/month/year) 30/05/2008
International Patent Classification (IPC) or both national classification and IPC G06Q10/00		
Applicant CLEAR CHANNEL UK LIMITED		

This International Searching Authority hereby declares, according to Article 17(2)(a), that **no international search report will be established** on the international application for the reasons indicated below

1. The subject matter of the international application relates to:
 - a. scientific theories
 - b. mathematical theories
 - c. plant varieties
 - d. animal varieties
 - e. essentially biological processes for the production of plants and animals, other than microbiological processes and the products of such processes
 - f. schemes, rules or methods of doing business
 - g. schemes, rules or methods of performing purely mental acts
 - h. schemes, rules or methods of playing games
 - i. methods for treatment of the human body by surgery or therapy
 - j. methods for treatment of the animal body by surgery or therapy
 - k. diagnostic methods practised on the human or animal body
 - l. mere presentations of information
 - m. computer programs for which this International Searching Authority is not equipped to search prior art
2. The failure of the following parts of the international application to comply with prescribed requirements prevents a meaningful search from being carried out:

<input type="checkbox"/> the description	<input checked="" type="checkbox"/> the claims	<input type="checkbox"/> the drawings
--	--	---------------------------------------
3. A meaningful search could not be carried out without the sequence listing; the applicant did not, within the prescribed time limit:
 - furnish a sequence listing on paper complying with the standard provided for in Annex C of the Administrative Instructions, and such listing was not available to the International Searching Authority in a form and manner acceptable to it.
 - furnish a sequence listing in electronic form complying with the standard provided for in Annex C of the Administrative Instructions, and such listing was not available to the International Searching Authority in a form and manner acceptable to it.
 - pay the required late furnishing fee for the furnishing of a sequence listing in response to an invitation under Rule 13ter.1(a) or (b).
4. A meaningful search could not be carried out without the tables related to the sequence listings; the applicant did not, within the prescribed time limit, furnish such tables in electronic form complying with the technical requirements provided for in Annex C-bis of the Administrative Instructions, and such tables were not available to the International Searching Authority in a form and manner acceptable to it.
5. Further comments: see further information sheet

Name and mailing address of the International Searching Authority 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 Katrin Sommermeyer
--	--

FURTHER INFORMATION CONTINUED FROM PCT/ISA/ 203

The claims relate to subject matter for which no search is required according to Rule 39 PCT. Given that the claims are formulated in terms of such subject matter or merely specify commonplace features relating to its technological implementation, the search examiner could not establish any technical problem which might potentially have required an inventive step to overcome. Hence it was not possible to carry out a meaningful search into the state of the art (Art. 17(2)(a)(i) and (ii) PCT; see PCT International Search Guidelines, Chapter 9).

The problems which are addressed do not appear to require a technical, but rather an administrative/organisational, or business, solution. The implementation of this solution may include the use of generic technical features, however these do not interact to solve any overall technical problem but merely serve their well-known functions.

The applicant's attention is drawn to the fact that claims relating to inventions in respect of which no international search report has been established need not be the subject of an international preliminary examination (Rule 66.1(e) PCT). The applicant is advised that the EPO policy when acting as an International Preliminary Examining Authority is normally not to carry out a preliminary examination on matter which has not been searched. This is the case irrespective of whether or not the claims are amended following receipt of the search report or during any Chapter II procedure. If the application proceeds into the regional phase before the EPO, the applicant is reminded that a search may be carried out during examination before the EPO (see EPO Guideline C-VI, 8.2), should the problems which led to the Article 17(2)PCT declaration be overcome.