



- (51) **International Patent Classification:**
G06T 19/00 (2011.01) *G06F 3/048* (2006.01)
- (21) **International Application Number:**
PCT/AU2012/000495
- (22) **International Filing Date:**
9 May 2012 (09.05.2012)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
61/486,164 13 May 2011 (13.05.2011) US
2011901836 13 May 2011 (13.05.2011) AU
- (71) **Applicant (for all designated States except US):** IGO-BUBBLE LIMITED [IE/IE]; Fitzwilton House, Wilton Place, Dublin 2 (IE).
- (72) **Inventors; and**
- (75) **Inventors/Applicants (for US only):** BURTON, Jade Wood [AU/DE]; Pappelallee 11, 10437 Berlin (DE). GALVIN, Kieran [IE/IE]; Fitzwilton House, Wilton Place, Dublin 2 (IE).
- (74) **Agent:** FREEHILLS PATENT & TRADE MARK ATTORNEYS; Level 43, 101 Collins Street, Melbourne, Victoria 3000 (AU).

(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, CL, 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, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, 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, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, 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, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— of inventorship (Rule 4.17(iv))

Published:

— with international search report (Art. 21(3))

(54) **Title:** METHOD IN A COMPUTING SYSTEM

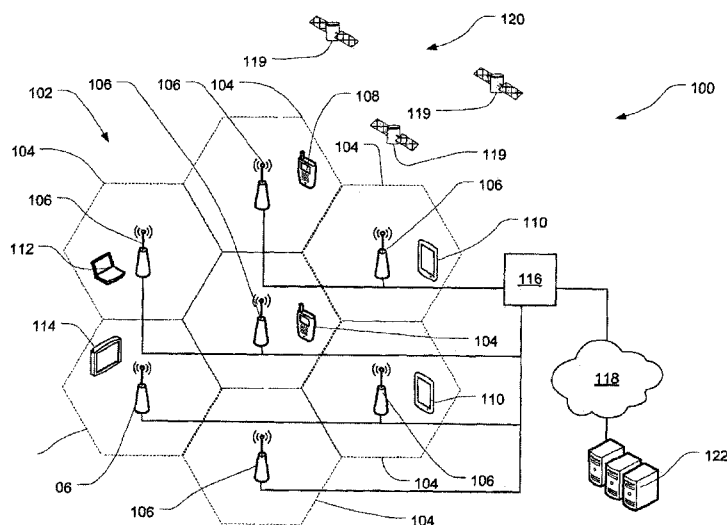


FIG. 1

(57) **Abstract:** The present invention relates to methods in a computing system, a computing system (122) and clients (212) for communicating with a computing system. The system includes a data storage system (200) storing data representing at least one virtual object; and a data processing system. The method includes identifying an interaction between a virtual object and a user; and updating virtual object behaviour data relating to one or more virtual objects in response to the identified interaction.

Method in a computing system

Field of the invention

The present invention relates to methods in a computing system, a computing system and clients for communicating with a computing system. In a preferred form the present invention
5 relates to an augmented reality system, components therefore, and methods in such a system.

Background of the invention

Augmented reality systems exist in which a user's device is provided with a hybrid visual, auditory or other interface including a first portion which represents reality and a second portion, usually overlaid on top of the first portion, to provide additional information than can be
10 perceived by the user in real time. For example, in some systems the mobile user terminal is used as a view finder through which a scene can be viewed and onto which is overlaid points marking virtual objects and locations that may be of interest to a user.

In augmented reality systems, virtual objects may exist. Virtual objects typically exist as data in a database structure of the augmented reality system which specifies a virtual object's
15 parameters such as its appearance and access permission. Typically, these parameters are fixed or only able to be changed by a virtual object creator or owner. This can lead to many virtual objects becoming stale over time and uninteresting to users as the virtual object's properties are essentially static.

However, augmented reality systems of this type are typically constrained in the way that
20 a user may interact with them and accordingly over time user interest in the systems may diminish. In a commercial sense, if these systems are used for notifying potential customers or otherwise advertising to users a decrease in user interest will translate into a decrease in advertiser interest and hence a decrease in revenue for the system operator.

Accordingly, it would be desirable to have an augmented reality system in which the
25 interaction of the virtual object with the user is more dynamic.

Reference to any prior art in the specification is not, and should not be taken as, an acknowledgment or any form of suggestion that this prior art forms part of the common general

knowledge in Australia or any other jurisdiction or that this prior art could reasonably be expected to be ascertained, understood and regarded as relevant by a person skilled in the art.

Summary of the invention

The present inventors have realised that augmented reality systems may be improved by
5 allowing augmented reality virtual objects to evolve over time. More particularly, the augmented reality virtual objects can preferably be adapted to change their behaviour in response to interactions with users of the system. Any aspect of the behaviour of a virtual object may change including, but not limited to, the way in which a virtual object behaves during
10 interactions with either users or other virtual objects, the rendering behaviour of the virtual object, or a health parameter of the virtual object. Triggers for changing the behaviour of the virtual object are preferably based on an interaction between a user and the virtual object, or the user and another virtual object. Interactions may take many forms and can include interactions in which the user is passive or active and even interactions in which the user ignores or fails to interact with the virtual object.

15 Advantageously, by updating data stored in the database upon user interaction, the virtual objects do not become stale over time, and they gain an animacy and interactivity not realisable in the more static prior art systems.

In one aspect, the present invention provides a method in a computing system. The computing system includes: a data storage system storing data representing at least one virtual
20 object, said data representing the at least one virtual object including virtual object behaviour data and virtual object location data, and a data processing system configured to process data stored in the system, and to communicate with a user device via at least one communications channel. In this case the method includes: identifying an interaction between a virtual object and a user; and updating virtual object behaviour data relating to one or more virtual objects in
25 response to the identified interaction.

The step of identifying an interaction can comprise a step of receiving data identifying a virtual object.

The step of identifying an interaction can comprise a step of receiving data representing a user input signifying a user initiated interaction with a virtual object.

The user initiated interaction can include any one of:

- the user using the virtual object;
- the user copying the virtual object;
- the user retaining the virtual object;
- 5 the user viewing the virtual object;
- the user accessing the virtual object;
- the user accepting the virtual object;
- the user capturing the virtual object;
- the user moving the virtual object;
- 10 the user modifying the virtual object; and
- the user releasing the virtual object.

In response to receiving the user initiated interaction, the computing system can perform a step of transmitting data including data enabling or facilitating the rendering or other reproduction of the virtual object.

- 15 The step of identifying an interaction can comprise: receiving data representing a location of a user; determining a proximity between a user and a location corresponding to the virtual object location data; and in the event that the proximity is within an predetermined radius of the virtual object and/or user, determining that an interaction has occurred.

- 20 The step of updating virtual object behaviour data relating to one or more virtual objects, can include updating association data relating the virtual object to the user.

The step of updating association data relating the virtual object to the user can include updating the virtual object location data for the virtual object.

The method can further include checking access permission data of at least one of the virtual objects or user.

In response to receiving the user initiated interaction, the computer system can perform a step of copying the virtual object.

The step of updating the association data can comprise associating the user with the virtual object. The step of updating the association data can comprise dissociating the user from
5 the virtual object.

Updating the virtual object behaviour data can include modifying the data representing the virtual object such that its location data is determined by reference to a user rather than a location.

Updating the virtual object behaviour data can include modifying the data representing
10 the virtual object such that its location data is determined by reference to a location rather than a user.

Modifying the virtual object behaviour data can include a step of modifying data corresponding to the rendering or other reproduction behaviour of the virtual object.

Modifying the virtual object behaviour data can include a step of modifying data
15 corresponding to the interaction behaviour of the virtual object.

Modifying the virtual object behaviour data can include a step of modifying data corresponding to the health of the virtual object.

Modifying the virtual object behaviour data can include a step of modifying data corresponding to a virtual object related to the virtual object.

20 Modifying the interaction behaviour data can include a step of updating or providing user permission to interact with the virtual object.

The step of modifying the interaction behaviour data can include a step of updating or providing an interaction type specifying the nature of interaction between a user and virtual object; or between virtual objects.

25 The step of modifying the interaction behaviour data can include a step of updating an interaction radius applicable to the virtual object.

The step of modifying data corresponding to the rendering behaviour or reproduction behaviour can include providing data representing a style of rendering the virtual object.

The step of modifying data corresponding to the rendering behaviour can include providing data representing a visibility radius applicable to the rendered virtual object.

- 5 The step of modifying data corresponding to the health of the virtual object can include updating data corresponding to a time to live of the virtual object.

The step of updating virtual object behaviour data can include updating data that indirectly moderates a behaviour of the virtual object.

- 10 The method can include incrementing, decrementing or re-setting a time to live of a virtual object in response to a user interaction with a virtual object.

- 15 The data storage system can also store data representing a health parameter of a virtual object. In this case the method can include: updating the health parameter of one or more virtual objects from time to time. The health parameter of one or more virtual objects can be updated periodically. The step of updating preferably includes decrementing the health of one or more virtual objects. The health parameter of substantially all virtual objects can be decremented periodically.

- 20 Preferably data relating to the virtual object involved in the identified interaction is updated. However, in some embodiments the step of updating virtual object behaviour data relating to one or more virtual objects can include updating data relating to at least one virtual object associated with the virtual object involved in the identified interaction. The virtual object(s) to be updated could be associated with the virtual object by any one or more of the following attributes:

- virtual object creator;
- virtual object antecedent or lineage;
- 25 • virtual object access or interaction history;
- virtual object owner;

- virtual object proximity;
- virtual object type; and
- a user/creator/owner defined association.

In a second aspect of the present invention, there is provided an augmented reality system
5 including: a data storage system. The data storage system stores data representing:

at least one virtual object, including virtual object behaviour data and virtual object
location data; and at least one user.

The system also includes data processing system configured to implement a method as
described herein. Most preferably the method is performed in accordance with an embodiment
10 of the first aspect of the invention.

The data storage system can include media storage for storing media virtual objects
associated with one or more virtual objects. The augmented reality system can further include at
least one interface to a communications network to receive and/or transmit data from and/or to a
user client device. The interface(s) can be adapted to exchange any one or more of: interaction
15 data, location data, virtual object data with the user device.

In another aspect, the present invention provides a client for an augmented reality system.
The client provides an augmented reality interface including, an user interface portion
corresponding to a real world scene and a portion representing one or more virtual objects; a
user input portion enabling a user to interact with a virtual object according to behaviour data
20 corresponding with a virtual object. The client also includes a communication portion confined
to communicate with an augmented reality system to exchange at least interaction data and
behaviour data.

The client can be configured to perform or enable any one or more of the following
interactions:

- 25 the user using the virtual object;
- the user copying the virtual object;

the user retaining the virtual object;
the user viewing the virtual object;
the user accessing the virtual object;
the user accepting the virtual object;
5 the user capturing the virtual object;
the user moving the virtual object;
the user modifying the virtual object; and
the user releasing the virtual object.

The client can be adapted to interact with a system as described herein. Most preferably it
10 is adapted to interact with a system according to the second aspect of the present invention.

The client can be implemented in a mobile computing device. The mobile computing device preferably includes a locator component configured to determine the location of the mobile computing device. Most preferably the client is a software application configured to be run on a computing device.

15 As used herein, except where the context requires otherwise, the term "comprise" and variations of the term, such as "comprising", "comprises" and "comprised", are not intended to exclude further additives, components, integers or steps.

Further aspects of the present invention and further embodiments of the aspects described in the preceding paragraphs will become apparent from the following description, given by way
20 of example and with reference to the accompanying drawings.

Brief description of the drawings

Embodiments of the present invention will now be described, by way of non-limiting example. In the drawings:

Figure 1 is a schematic representation of a communications, computing and positioning
25 system infrastructure in which a system according to the present invention may be implemented;

Figure 2A illustrates schematically an augmented reality system and client application used in an embodiment of the present invention;

Figure 2B and 2C together provide an entity relationship diagram illustrating an exemplary set of database tables used in an embodiment of the present invention;

5 Figure 3A is a flowchart illustrating the process of determining user and virtual object position within the augmented reality system according to one embodiment of the present invention;

Figure 3B illustrates the movement of a user through space illustrating virtual objects moving into range and out of range in the example of figure 3A;

10 Figure 4 is a sequence diagram illustrating an exemplary user initiated interaction, in the form of a request to view a virtual object, according to an embodiment of the present invention;

Figure 5 illustrates a flowchart illustrating how a user interaction may cause a virtual object may be split in an embodiment of the present invention; and

15 Figure 6 illustrates a flowchart showing an example of an interaction between a user and a virtual object, where the interaction is initiated by a virtual object without user input.

Detailed description of the embodiments

Figure 1 is a schematic representation of a communications, computing and positioning system infrastructure in which a system according to the present invention may be implemented. The infrastructure 100 generally includes a network 102, positioning system 120, and augmented reality system 122. The network 102 will typically be a telecommunications system such as a cellular telephone and data network. However, as will be appreciated by those skilled in the art the network 102 need not be a cellular telephone network, as is illustrated, but may be any type of communication network capable of communicating with user devices over a data channel.

25 The cellular telephone and data network 102 illustrated includes a plurality of cells 104, having has a respective base station 106 for communicating with user devices located in a surrounding area. A plurality of user devices are in communication with the network 102. In this example the user device are in the form of, smart phones 108, tablet computer devices 110, notebook computer 112 and mobile computing device 114, the mobile computing device may for

instance be a navigation system or other in-car computing system. As will be appreciated, many other types of user device could be used in embodiments of the present invention and the invention should not be considered as being limited to the examples given herein.

The transmitters 106 of the communications network 102 are communicatively connected to a core or backbone network which enables communication between the various cells, and with other networks via a network control infrastructure subsystem, e.g. 116. The network 102 is also connected to, or forms part of, a data communications and computing network 118, such as the Internet. The user devices 108 to 114 operating within the telecommunications network 102 preferably also receive signals from a constellation of positioning satellites, such as the positioning satellites 119 of the GPS system 120, in order to determine their respective position. The positioning may be performed using Assisted-GPS whereby part of the GPS data is delivered via the communications network 102 to the user device 108 to 114. The user devices 108 to 114 communicate with an augmented reality system 122 via the telecommunications network 102 and data communication network 118.

Figure 2A illustrates more details of an embodiment of the augmented reality system of the type illustrated in figure 1. The system 200 shown in Figure 2A, communicates with a user device running a client application 212. The system 200 includes the following main subsystems:

- 1) A data storage subsystem, including a database 201 and cloud storage 204.
- 2) An Application Server 202 (or group of servers) and push notification service 206, that mediate communication between the client application 212 and the data storage system.
- 3) A web interface comprising a web service 208 and dynamic website 210.

The client application 212 provides an interface through which a user interacts with the system 200. User can also interact with the system 200 using a web browser 214 to access the dynamic website 210.

The functional role of each of these subsystems will now be described in more detail.

Database storage subsystem

The database 201 includes one or more tables 200.1, and associated stored procedures 200.2. The database 201 stores the information that is required to model of the state of the virtual objects in the augmented reality world.

- 5 The database 201 can store data representing all entities within the augmented reality world, the primary entities being virtual objects and users. However other entities may also exist, e.g. virtual object owners, such as advertisers or corporations that are not users, system administrators or management. The database 201 can also store system business rules and other data necessary for the system to operate.
- 10 Figures 2B and 2C together present an entity relationship diagram illustrating an exemplary set of database tables that can be used in an embodiment of the present invention. Tables 1 to 3 set out attributes that can be stored in the entity tables for virtual objects, users and modules respectively, in an embodiment of the present invention.

Entity table definitions

Attribute	Description
EntityID	Unique Surrogate identifier for the virtual object.
Version	Row version for optimistic locking and caching.
GpsCoordinate	GPS coordinate of the virtual object.
CreationDateTime	Date and time the object was created.
EntityDescription	Optional text description of the virtual object.
CanvasID	ID referring to the Canvas of the virtual object.
Flags	Bit flags indicating extra options.
Radius	Radius of the virtual object, in metres.
CreatorAccountID	ID referring to the Account responsible for creating the virtual object.
EntityTypeID	An INT indicating how to interpret the row, i.e. 1 = Viral Bubble.
Size	The health of the virtual object, if applicable.
HolderAccountID	ID referring to the Account of the user who presently holds

	this virtual object. May be NULL if the virtual object is not held by anyone. Mutually exclusive to GpsCoordinate in some embodiments.
FamilyID	A number that acts like a logical “last name” of the virtual object. A virtual object has the same FamilyID as the virtual object that it was cloned from or split from. In this way, virtual objects can be grouped by a common FamilyID.
DistanceTravelled	A number that is incremented whenever the virtual object’s GpsCoordinate changes. This is a cumulative distance, in metres.
ViewCount	A number that is incremented whenever the virtual object’s canvas is viewed by user.
EditCount	A number that is incremented whenever the virtual object’s canvas is modified by a user.
CaptureCount	A number that is incremented whenever the virtual object is captured.
BirthplaceGpsCoordinate	A GPS coordinate that is set once, when the virtual object is cloned at a location, or splits at a location, or is initially released at a location.

Table 1: Attributes stored for virtual objects in an exemplary embodiment of the present invention

Account table definitions

Attribute	Description
AccountID	Unique Surrogate identifier for a user.
Version	Row version for optimistic locking and caching.
ProfileCanvasID	ID referring to a canvas that represents the user’s profile.
Username	A friendly username, chosen by the user.
PrivateEmailAddress	If the user forgets their password, they can opt for it to be sent to this email address automatically.
Password	The user’s secret password, used for authentication.
CreationDateTime	The date and time that the account was created.

IsOnline	A 1 or 0 value indicating that the user is presently online. Speeds up queries because they can often ignore offline users.
LastLoggedOff	The date and time that the user last logged off.
DeviceToken	A device token used by Apple Push Notification Service to identify the user's device.
LastUpdated	The date and time that the account was last modified.
LastLoggedIn	The date and time that the user last logged on.
GpsCoordinate	A last-known GPS coordinate of the user.
ChangePassword	A password to be used the next time the user logs on. This allows the user to continue using their old password until they log out. When not used, this field is NULL.

Table 2: Attributes stored for a user in an exemplary embodiment of the present invention

Module table definitions

Attribute	Description
ModuleID	Unique Surrogate identifier for a module.
Version	Row version for optimistic locking and caching.
ModuleTypeID	An INT indicating how to interpret the row, i.e. 14 = Picture Module.
EditAccess	An enumeration indicating what general edit access users have.
ReviewStatus	An enumeration indicating whether this object has been reviewed by moderators, and if so, what the outcome of the review was.
RejectReason	A string field which may contain a free-text explanation or comment on the ReviewStatus, as entered by a moderator.
CreationDateTime	The date and time that this module was created.
DefaultWidth	A module may appear as many different sizes on multiple canvases simultaneously. This field indicates what width to use when the object is first added to a canvas.
DefaultHeight	See DefaultWidth.

Picture_Caption	For Picture modules, this is an optional caption.
Picture_DataObjectID	For Picture modules, refers to a file stored in the cloud which contains the picture data.
TapToCall_PhoneNumber	For "Tap to call" modules, the phone number to dial when the user taps on the button.
Video_StillImageObjectID	For Video modules, refers to a file stored in the cloud which contains a still image of the video.
Video_DataObjectID	For Video modules, refers to a file stored in the cloud which contains the video data.
Url_Url	For URL modules, the URL to browse to when the user taps on the button.
GpsLocation_GpsPosition	For GPS Location modules, the GPS position to reveal on a map.
CalendarReminder_DateTime	For Calendar Reminder modules, the date/time to add to the user's calendar when the user taps on the button.
CalendarReminder_Description	For Calendar Reminder modules, a text description of the reminder.
BusinessCard_PhoneNumber	For Business Card modules, a phone number to add to the user's contacts list when the user taps on the button.
BusinessCard_EmailAddress	For Business Card modules, an email to add to the user's contacts list when the user taps on the button.
BusinessCard_Title	For Business Card modules, the title of the business.
BusinessCard_Address	For Business Card modules, the address of the business.
File_DataObjectID	For File modules, refers to a file stored in the cloud.
Text_Text	For Text modules, refers to the text that will be displayed.
Text_FontSize	For Text modules, indicates the size of the font to use.
Text_FontName	For Text modules, indicates the font name to use.
Text_Red	For Text modules, indicates the red component of an RGB colour to use.
Text_Green	See Text_Red, Text_Blue.
Text_Blue	See Text_Red, Text_Green.
Canvas_Width	For Canvas modules, indicates the width of the canvas in

	pixels. Note that a Canvas is itself a Module, and so resides in the same table.
Canvas_Height	For Canvas modules, indicates the height of the canvas in pixels.
Canvas_BackgroundRed	For Canvas modules, indicates the red component of the RGB background colour.
Canvas_BackgroundGreen	See Canvas_BackgroundRed, Canvas_BackgroundBlue.
Canvas_BackgroundBlue	See Canvas_BackgroundRed, Canvas_BackgroundGreen.
IsTemplate	1 if this is a Canvas module that is to be usable as a template. When the user creates a new virtual object, a list of template canvases will be presented to act as the “starting point” for the new object.

Table 3: Attributes stored for Modules in an exemplary embodiment of the present invention.

The database 201 can be centralised or split across several databases (as illustrated in this embodiment) which are stored on one or more servers and/or server farms.

- 5 The system may store data remotely on a cloud storage server 204. The cloud storage system is used to store media and other data that forms part of the virtual objects. This data is stored in a cloud storage system so as to minimise data transfer from the main database 201 which stores system state. This can speed up system access and allow system expansion as more virtual objects are created over time. Clearly choice between centralised and cloud storage or the
- 10 balance between them can change depending on system requirements.

Virtual objects can contain a range of information. Turning firstly to their appearance in the virtual world and the user experience they deliver to users that interact with them in use.

- Virtual objects can be viewed in a similar fashion to websites or other documents, in that they have a structure defined by a set of parameters, termed a “canvas” herein, and content, termed
- 15 “modules”. Thus a virtual object can comprise one or more “modules” arranged on a “canvas”.

A canvas in one form has a height and width, measured in pixels; although non visual virtual objects could also be created, in which case other parameters defining their structure will be used, say duration etc.

A *module* is a piece of digital media that can be distributed with a virtual object, e.g. by placement on a canvas, or that is made accessible at a location specified with a fixed virtual object. Examples of modules include but are not limited to, the following:

- A text block, having a font, font size, colour, rotation and alignment.
- 5 • An image, e.g. possibly having a rotation and a non-destructive cropping region, so that it can be changed many times without affecting the underlying pixel data. The cropping region could be any shape or weight, allowing the user to paste on non-rectangular “cut outs” of people and virtual objects.
- 10 • A video, including parameters as to whether it is to be played embedded or full-screen.
- A sound recording.
- A Location, e.g. GPS location.
- A Calendar Event, invitation etc.
- A Document, e.g. in portable document format or any other document format.
- 15 • A shared file.
- A hyperlink.
- A telephone number.
- An email address.
- A business card.
- 20 • An RSVP.

A virtual object may have a different appearance depending on whether a user is merely viewing it from a distance or has captured it. Typically a virtual object's appearance or modes of interaction will be much richer to a user that possesses the virtual object, but present a less sophisticated or detailed interface to users passively observing it or passing by. For example, the

virtual object can have a thumbnail image that is created by cropping a main image of the virtual object or similar or play a snippet of a video or sound rather than the whole thing.

In addition to the content of virtual object that can be experienced by a user during an interaction, virtual objects also include a range of other data, including but not limited to:

- 5 Lineage data and ownership data – this data is used to record and track the creation and manipulation of virtual objects over its life. Some aspects can be used to modulate interactions with other virtual objects or users. This data can be data relating to one or more of the following:

Creator.

Owner.

- 10 Related virtual object data, family data.

Virtual object data relating to parent or child virtual objects.

Authentication or password data.

Location and movement data – an important aspect of many virtual objects is their position and ability to move. Thus each virtual object will have at least one item of location data associated

- 15 with it. Location data can include, but is not limited to:

Current position.

Movement status, e.g. whether the virtual objection is fixed, mobile, autonomous etc.

Movement parameters, such as speed, direction, conditions upon movement.

Target position, destination.

- 20 Historical position, originating position.

Health data – as will become apparent from the following description a virtual object's behaviour within the virtual world can be modulated by its health. Thus each virtual object will have some health related data, for example:

A virtual object size.

A time to live.

A status, e.g. live, dead, expired, etc.

Interaction data – interaction between virtual objects and users is central to the operation of the Augmented reality system according to the present invention. In addition to the other data stored
5 in the database, which may affect how a virtual object interacts with users or other virtual objects, some interaction-specific data can also be stored in the database, for example:

Interaction radius.

Visibility radius.

Interaction permission data.

10 Interaction types.

Data recording historical interactions.

Business and analytic data - This data can be used by system administrators, virtual object creators or owners, advertisers or others to monitor virtual objects. For instance, the data stored could include:

15 Owner.

Billing details for commercial virtual objects.

Number of users in-range.

Number of times viewed.

Number of times connected-to.

20 Average views per day.

Number of times Liked.

Number of times edited.

Comments left by users.

Average repeat visits per user.

Number of virtual objects in family.

Total distance travelled.

Status: alive or expired.

5 Number of times released.

Number of times captured.

Number of times cloned.

Data related to users is also stored in the database tables. This data represents the state of user in the augmented reality system, and can include data relating to:

10 User Identity. A wide range of user identity data could be stored, including, but not limited to:

Username.

Name.

Contact details (address, email, telephone, fax etc.).

Age or date of birth.

15 User demographic data.

Other user data, e.g. interests, club memberships etc.

Permissions and Virtual object interactions – A user can have different permissions for interaction with certain virtual objects based on a range of characteristics, for example whether they are a system manager, system user, merchant, advertiser, member of a group, e.g. a club or professional organisation, child, etc. Accordingly the database can store data, including but not limited to any one of the following types of permission or interaction data (either storing permissions or exclusions) for a user:

20

User type.

User creation data.

User age group.

Organisation membership data.

Permitted interaction types.

- 5 Virtual object data – each time a user interacts with a virtual object, data relating to the interaction can be stored for the user, for example:

Virtual object identification data, including, currently possessed virtual objects and/or recently possessed virtual objects, or

Interaction parameters for virtual objects including time, date, type, other parameters

- 10 User location data – the system needs to know where a user is in order to allow it to determine how or if the user can interact with virtual objects in the augmented reality world. Accordingly the database 200 can store data such as:

Current location.

Previous location(s).

- 15 Trajectory.

Speed.

Client Application 212

- The client application 212 provides an interface for stimulating the user and allowing the user to interact with virtual objects as part of the augmented reality service in real time. The client application possesses interfaces that can allow a user to interact with virtual objects, for instance by allowing the user to view, hear, access, copy, modify, capture, accept, retain, move, release or share the virtual object. The client application can also be adapted to allow the user to create, clone or edit virtual objects.
- 20

The client application 212 will typically be run on a mobile user device, such as a mobile phone, tablet computer or other user computing device. In one example, the device running the client application could be an Apple iPhone running iOS.

The client application 212 has a communication component 212.1 for interfacing with the application server 202. The communications component communicates with the application server 202 using a messaging protocol. The messaging protocol is preferably a connectionless protocol, for example, the protocol is the User Datagram Protocol (UDP), delivered over the Internet Protocol (IP). The client application 212 also includes an interface layer 212.2 that controls the interface through which the user interacts with the augmented reality system. The UI may have various components, e.g. a graphical user interface, audio interfaces such as sound receiving and output components, tactile interfaces e.g. a motion sensitive input device such as an accelerometer and vibrating output devices. It is convenient to describe a visual user interface and virtual objects that will have a visual aspect to them, however it should be noted that the user interface need not be visual in nature, and virtual object need not be perceived visually by a user. Any type of interface could be used which is perceived by any of a user's senses. Clearly the interface can use multiple modes of interaction which are perceived in different ways – in fact many interactions will use a hybrid interface, including a combination of visible, auditory and/or tactile stimuli.

The client application 212 also transmits location data of the user device to the application server 202. It does this by obtaining a GPS coordinate from the mobile device's built-in GPS chip, which is in communication with the positioning system 120. The client application 212 then takes the GPS coordinate and sends it to the Application Server 202 in a "Report GPS Position" request. This message is sent whenever a new GPS coordinate is received from the built-in GPS device, which can happen frequently, particularly if the user is moving.

The mobile client 212 receives messages from the Application Server 202 containing information relating to the virtual objects stored in the database 200, and in response provides outputs via the user's device to form an augmented reality interface. In use, the mobile client application 212 displays nearby and in-range virtual objects in relation to the user.

In one mode of operation the user interface 212.2 of the mobile client operates in a Finder mode. The Finder mode presents an interface displaying in-range virtual objects that the user

can currently interact with. The interface might, for example be a simple list interface, or provide a more dynamic graphical interface, e.g. a window full of bubble-like icons representing in-range virtual objects that float around on the screen. The icons can be represented as bouncing off each other and the walls in a realistic way. The user can touch and drag the bubbles on the screen to interact with them in the manners described herein. Similar interfaces can also be used to display virtual objects that the user has captured, or which have attached themselves to the user. Other interfaces are possible.

Other interfaces (or interface elements) can be provided to help a user to find an object which the user is able to see (or detect), but that is not close enough to interact with. Such an interface might look like a radar display showing icons for nearby virtual objects. The interface can also include a navigational aid, say, resembling a compass to guide a user to virtual objects.

In addition to showing in-range virtual objects, the Finder mode also helps the user to discover virtual objects that are nearby but technically not in-range. Indicators are shown beneath the list items that represent in-range virtual objects, and these each display a bearing, represented as a small compass, and a minimum distance the user must travel for the virtual object to be in-range.

The mobile client application 212, when running on a mobile device that is physically located at, or near a physical location that corresponds to a virtual object's location can display the canvas including any modules on the screen at the request of the user. The mobile client application 212 can present the canvas to the user based upon the position and orientation of the mobile device.

In particular, the mobile device can display a picture that comprises a representation of what the user would have seen when looking in that particular direction, and overlays upon that picture one or more modules from nearby virtual objects.

Access to the system may not always be provided by a dedicated client application. Instead some or all of the functionality of the client application could be provide as a features in an application with a different primary use. For instance, an application that takes photos and applies filters to them could include a "Place virtual object here" button, which uses an appropriate library to add a photo to the system server, as a virtual object. This way, the system can receive content from third party applications without users needing to download a full copy of the client application. As will be appreciated, such a feature could be added in a new version of a user's favourite Photo-taking application as a minor update.

Application Server 202

The database 201 is connected to an Application Server 202. The Application Server 202 performs two main roles.

In the first role, the Application server mediates communication between a mobile client 212 and the database 201. The communication can take a wide variety of forms. The following description will illustrate the nature of communications between the application server 202 and mobile client 212. By way of example the application server 202 can communicate with a mobile client 212 to authenticate a user's identity or client's identify in order to allow access to virtual objects stored in the database; and/or convey information, such as data for representing the virtual objects visually, to the mobile client application. The Application Server may communicate with the mobile client 212 via an intermediary, for example, via a push notification server 206.

In the second role, the Application server 202 can update the database 201 according to business rules. For example, each virtual object can have an associated health parameter that is stored within the database. According to a business rule, the health parameter is decremented at regular intervals, and any virtual objects which have a health parameter that is less than a predetermined value are updated to ensure that they do not appear in the augmented reality display. In an alternative embodiment, a virtual object expiry timestamp is consulted and/or updated as each virtual object is accessed.

The Application Server 202 uses a message bus architecture that allows for asynchronous unsolicited messages, or Notifications, to be dispatched to all relevant connected clients. For instance, whenever a virtual object is captured or released, the Application Server 202 can automatically transmit a Notification to all connected clients who are in-range of the virtual object. The Notification contains a copy of the virtual object, and other information such as which user performed the capture/release.

The Application Server 202 only selects, inserts, updates and deletes rows in the database 200 via stored procedures 200.2. This provides a layer of abstraction between the database schema and the Application Server 202 that reduces or eliminates the changes necessary on the Application Server 202 if the underlying schema of the database 201 changes, and vice-versa. The stored procedure layer also potentially affords a higher degree of security.

When the Application Server 202 creates the virtual object or fixed virtual object, it inserts a row into the Entity table. An example Entity table is illustrated in Figure 2B. Entity is an abstraction of any virtual object that may have a GPS coordinate and which may be spatially queried. The Entity table has a Spatial Index, which is a feature of Microsoft SQL Server. This, 5 makes it possible to quickly perform a query of "every virtual object within 100 meters of X, Y" etc. Spatial queries are used heavily in the system.

The Application Server 202 could run Microsoft Windows 7 or 2003 (or other operating system), and hosts a daemon (a.k.a. "Windows Service") written in C#.NET, which responds to requests from client application 212 apps.

10 **Web browser 214, dynamic website 210 and web services interface 208**

Virtual object information from the database 201 may also be accessed via a web browser 214 that is running on a user's or an administrator's computing device. The browser 214, such as Internet Explorer, Firefox or Safari, is used to access a dynamic website. The dynamic website 210 is constructed using a series of dynamically generated pages, authored in PHP and served 15 using the Apache webserver 210.1.

The webserver 210.1 may access the database 201 of virtual objects via a web services interface 208. In an alternative embodiment, the database 201 is accessed directly by the PHP code. In either alternative, the user is using the web browser and is able to access and/or update virtual object information.

20 In the embodiment, the web browser interface allows a merchant or advertiser to create a virtual object, and/or track the virtual object use and interactions over time. The merchant can view, for example, how many people interacted with the virtual object, the demographics of those people, and the co-ordinates through which the virtual object travelled over its lifetime. For example a merchant having a virtual object in the system can have access to an analytics portal accessible 25 to them via the website via which they can access the following information, which can be stored or derived from data:

Number of users in-range.

Location.

Virtual object birthplace.

Number of times viewed.

Number of times connected-to.

Average views per day.

5 Number of times Liked.

Number of times edited.

Comments left by users.

Average repeat visits per user.

Number of virtual objects in family.

10 Total distance travelled.

Status: alive or popped.

Number of times released.

Number of times captured.

Number of times cloned-from.

15 Demographic data for object host users.

Aggregated statistics over a range of virtual objects owner by the merchant can also be presented. Additionally, financial and account data is also accessible, for example historical transaction data, as well as a merchant's credit remaining for creating or maintaining their virtual objects.

20 Data can be presented in the form of reports, mapped geographically, temporally, graphed etc. For example, the number of times a given virtual object is viewed each day, or total views for all virtual objects each day, the path followed by a virtual object.

Virtual object behaviours and interactions

As noted above, as a user, possessing a user device running a client application 212,
25 moves around the real world they are able to interact with virtual objects in the augmented reality world that are overlaid onto it.

In most cases this requires the augmented reality system to determine the position of a user, and determine nearby virtual objects with which the user may interact. Figure 3A illustrates a method of performing this process. The left pane of Figure 3A relates to operations that take place in the mobile client, and the right pane relates to operations that take place in the database or the Application Server. Initially at 302, the mobile client determines the current GPS co-
5 ordinate of its mobile device. As noted above, user position is determined by the client application exchanging location data from the user device's inbuilt positioning chip with the application server. The mobile client transmits 304 the GPS co-ordinate to the database in a "Report GPS Position" message, which receives the co-ordinate 306.

10 In step 308, the database is consulted by the application server to identify any virtual objects that are near to the GPS co-ordinate of the mobile device. Specifically the Application Server 202, upon receiving the "Report GPS Position" message, performs a spatial query on the Entity table, selecting all rows whose GPS coordinate is within range of the user's current location. This range, typically expressed in meters, is typically defined by the radius property of each
15 virtual object, as stored in the Radius column of the Entity table. When the Application Server 202 has obtained a list of nearby Entities, it saves these in a second table, AccountInRangeEntity in step 310. This table creates a relationship between users (represented by the Account table) and Entities. For instance, if a user is in-range of 10 Entities, then there will be exactly 10 records in the AccountInRangeEntity table, respectively linking the 10 Entities back to the
20 Account of the user.

In step 312, information allowing visual display of virtual objects nearby to the GPS co-ordinate of the mobile device is then transmitted back to the mobile device.

In step 314, the mobile device receives the virtual object information, to generate a suitable UI display based upon the information. Subsequently, the process is essentially repeated, such that
25 the virtual object information is updated to account for any change in the user's location. That is, the user device updates its location in step 316, and sends the new location data 318 to the application server. The application sever 202 receives the updated location data at 320 and repeats its database spatial query 322. For instance, as a user moves across a city, new virtual objects and fixed virtual objects "come in range", while others "go out of range". The
30 Application Server 202 determines the changes in the in-range and out-of range virtual objects at 324 and reports these two events back to the client application 212 asynchronously via an event

feed 326, allowing the client application to instantly notify the user of newly discovered virtual objects, while also updating the display to remove no-longer-in-range virtual objects. This updating process is repeated 330 as the user moves. Figure 3B illustrates the movement of a user through space. The user is initially located at the centre of circle 350. The positions of virtual objects are indicated by dots. Initially the table of in-range virtual objects created by the application server includes all virtual objects within circle 350. As the user moves to the position at the centre of circle 352, the virtual objects in region 354 come into range, and those in region 356 go out-of-range. The application server reports the new in-range virtual objects and new out of range virtual objects contained in regions 354 and 356 (respectively) to the client application via an event feed.

Depending on the nature of the virtual objects within range of a user the mere presence of the user may cause an interaction to be detected, alternatively a virtual object could be determined if the user initiates an interaction. This difference is essentially one of whether an interaction is virtual object initiated or user initiated.

Some virtual objects will automatically form associations with users. For example an interaction may result from a user moving to a particular geographic position e.g. a position that corresponds to the present position of a virtual object. In such an example, when the user's location is detected to be within a particular zone, say within a particular interaction radius from a virtual object, the system determines that an interaction between the virtual object and that user has occurred. Responsive to the interaction, the system may initiate some action on the mobile client to attract the user's attention – for example, vibrating the mobile device, or playing an audible chime.

An interaction may also result from a user selecting an input on their mobile device. For example, a user may select an option from a menu that allows the user to select a type of interaction with a virtual object. For example, the user may choose to capture the virtual object – that is, take the virtual object from its location and carry the virtual object with them. The user may then subsequently release the virtual object at a different location, or trade the virtual object with another user.

Figure 4 illustrates a process for a user-initiated interaction. In this case the interaction is a user viewing a virtual object in detail. Figure 4 is divided into two panes – left and right. The left

pane 402 relates to operations that take place in the mobile client. The centre pane 403 relates to operations that take place in the database 201. The right pane 404 relates to operations that take place in the cloud storage service of the data storage system. Arrows between the panes indicate that a message is sent between the components.

- 5 As illustrated in Figure 4, initially at 406, the mobile client determines input from the user of the device that the user wishes to view one of the in-range virtual objects. In order to view a particular virtual object, mobile client application 212 issues a request to the server 408, that identified the virtual object to be viewed such as a "Get Canvas" request. In response to receiving the request at 410, the Application Server 202 queries the database 201 in step 412 and
- 10 selects the Canvas row from the Module table, plus all child Modules, aggregates these and returns (in step 416) the complete Canvas virtual object to the client application 212. Additionally the application server updates the virtual object's data in the database 201 to reflect that it has been requested by a user. For example the virtual object's time to live or health can be incremented according to the system's business rules, or virtual object's specific properties, thus
- 15 extending the virtual object's life, other data may also or alternatively be updated as will become apparent in the following examples. The client application receives the virtual object canvas information that in this case, omits the necessary media for the user device to render or otherwise reproduce the virtual object in a manner in which it can be perceived by a user, at 418. The client application then requests media content that is needed to recreate the virtual object from media
- 20 storage system, e.g. cloud storage 204 of figure 2, in step 420. The request is received at 422 and the storage system accessed in 424 and the required content returned to the client application in step 426. The client application receives the content of the virtual object 212 at 428 then performs layout and presents the Canvas at 430. As can be seen, in this example images, video and audio data contained in Modules are not stored on the Application Server 202. Instead, these
- 25 data are stored on and served by in cloud storage network 204. This means a PictureModule has a row in the Module table, but the actual pixel data are stored separately, in the cloud. The PictureModule thus has a Picture_DataVirtual objectID column that identifies the data virtual object in the cloud storage system 204 and allows it to be retrieved. Other embodiments can store the media module data to be stored in the main database 201.
- 30 Broadly speaking, step 414 of updating virtual object behaviour data in Figure 4 by increasing a virtual object's health, is concerned with ensuring that virtual objects which are most often

viewed are promoted within the system to allow them to be viewed more, whereas virtual objects which are rarely viewed will not continue to exist.

Specifically, each virtual object has an associated 'health' parameter, which is stored in the database. The health parameter is progressively reduced over time by interaction of the decay daemon and stored procedures 200.2 of the database 201, or through certain types of interaction. Health can also be increased by certain other interactions. The health parameter can moderate a virtual object's interaction, by allowing the system to keep a measure of the regularity and quantity of views. The concept of health can be used to promote scarcity in the virtual world by allowing old or less successful virtual objects to expire, thereby avoiding the space from becoming cluttered.

The idea of incrementing a health parameter upon an interaction can be extended by using the health parameter to cause the creation of a new virtual object. Figure 5 illustrates an example of this process, and can be considered to follow on from (or be a sub-process within step 414 of figure 4). In this regard, in an initial step 502 the virtual object's health parameter is incremented as in step 414 of figure 4. Next in 504 the virtual object's health parameter is compared to a predetermined threshold value. If the health parameter is greater than the threshold two things occur. First at 506, a new virtual object is created. The new virtual object is created having a health parameter that is approximately half of the original virtual object. Next, at 508 the existing virtual object health parameter is reduced to half its former level. This can be viewed as splitting the original virtual object into the two virtual objects. In this way, the process mimics the biological process of cell division – each virtual object taking part of the total 'health' created by views. In the case that, at step 504, the health parameter is not greater than the threshold, no further action is taken to update the virtual object behaviour data, and the virtual object continues with its newly incremented parameter. This type of behaviour allows popular virtual objects, as evidenced by its high health, to split and thereby be accessed by multiple users simultaneously.

The process of creating the new virtual object, can be performed as follows:

- 1) A *shallow* copy of the Canvas associated with the original virtual object is made.

- 2) A *deep* copy of the Entity row that represents the original virtual object is made.
- 3) The copied Canvas with the copied Entity row forms the new virtual object.

In this way, the canvas of the new virtual object still refers to the *same* Modules that the original Canvas. However, the position data and other relevant data associated with each Module in relation to its parent Canvas is copied and linked with the new Canvas, allowing the new virtual object to have different position information from the original virtual object.

It will be appreciated that after a split occurs, a new virtual object is added to the virtual world. Accordingly, the system performs a spatial query on the database 201 to determine all the users who are “in range” of the virtual object, before dispatching Notifications to them. These Notifications are interpreted by the client application 212 and the display is asynchronously updated to reflect the newly added virtual object.

The client application 212 is therefore responsible for downloading image, video and audio data from a separate server. It initiates this downloading as soon as the Canvas virtual object is retrieved from the Application Server 202. The downloading occurs using standard HTTP GET requests, and several downloads may occur simultaneously.

The data virtual objects fetched from the cloud storage 204 are preferably immutable; that is, they are never updated. This allows for aggressive caching on the client side, and prevents the same data virtual object from needing to be downloaded more than once.

In some embodiments of the system attribution of the original creators of particular modules is tracked – in particular, the system tracks the linkage between modules of content and its creator, such that even when content is mashed-up and recycled to produce derivative works, attribution of content is still possible.

Accordingly, if a user “clones” another's virtual object, the system will not allow the cloner to take credit for the original creator's work. The present system uses two separate mechanisms to achieve this:

- Maintaining metadata that relates each module to the author across copying and editing operations; and

- Comparing modules of content in newly created virtual objects to existing virtual objects, to determine whether each module is similar to an existing virtual object, in which case the content is attributed to the user.

It is noted that the 'health' parameter may be artificially inflated by a single user repeatedly
5 viewing a particular virtual object. To protect against users artificially inflating the health parameter for a virtual object, only the first view by a particular user within a predetermined time period may be used to determine the health of a virtual object.

Capturing virtual objects

Figure 6 illustrates a flowchart showing an example of an interaction between a user and
10 a virtual object, where the interaction is initiated by a virtual object without user input. Figure 6 follows a similar convention as for Figure 4, whereby the left pane 602 relates to operations that take place in the mobile client and the right pane 604 relates to operations that take place in the application server. In the interaction being described, the virtual object can be considered a viral object, which unilaterally attaches itself to a user as a user moves into range.

15 In an initial step 606 the client application retrieves the user's device location from the positioning chip of the user device. Next in 608 the user location is transmitted to the application server. The application server receives the user's location at 610 and performs a spatial query on the database to determine all virtual objects within a predetermined range. Next, 612, if the virtual object has suitable interaction behaviours, the application server determines the user is
20 within an interaction radius for the virtual object. The database identifies that the virtual object and user will interact and updates the virtual object's behaviour data 614 to reflect that the virtual object is now associated with the user. Conceptually this interaction can be seen as the object *hitchhiking* on a user by jumping on the user, and then being carried around by the user (until it is dropped or otherwise moved on). In this case the behaviour data that is updated is the
25 location data of the virtual object. More specifically, when a virtual object *hitchhikes* or (a user chooses to "capture" a virtual object), a row in the Entity table corresponding to that virtual object is updated. The update sets the GpsCoordinate column to NULL and sets the VirtualObjectAssociatedWithAccountID column to the ID of the user's Account. Thus, instead of the virtual object having a location, it has its location determined by association with a user. If the
30 virtual object was stationary the virtual object will now become mobile and move with its user.

The user's account is also updated reflect that the virtual object is held by the user in 618, and the user is notified of his or her new virtual object at 620. In a similar fashion to the example illustrated in Figure 4, the health parameter of the virtual object may also be updated e.g. to make it healthier, increasing its time to live. Information relating to one or more virtual objects is then sent back to the user's device 614 where it is received 616 and may be displayed 618.

Virtual objects can have different behaviours depending on whether they have been captured by the particular user. By way of example, the act of capturing a virtual object may be necessary prior to allowing a particular user to view the canvas of the virtual object, or a virtual object may not be visible or discoverable by other users if it is possessed by a user. The variation in behaviour of a virtual object can be controlled by associating a virtual object with a particular user. For example, a private message may be sent from one user to another user by restricting the users able to view the message.

Alternatives to this situation can be implemented, for instance capturing a virtual object can be dependent on a user input. Moreover, for a virtual object that is free to hitchhike on a user, the user may be able to reject the virtual object. This can be viewed as a negative interaction and user to decrement the virtual object's health parameter.

Releasing a virtual object is essentially the opposite process to capture. In essence the process of releasing a virtual object involves disassociating a particular user from the virtual object, and updating the virtual object to have a particular set of GPS co-ordinates. Specifically, using the example given above, releasing a virtual object involves setting VirtualObjectAssociatedWithAccountID to NULL and GpsCoordinate to the present GPS coordinate of the user.

Like the capture process, users release virtual objects for a variety of reasons, for example:

- The user might be bored with the content of the virtual object. Maybe he has viewed the video a few times, or read the joke, or browsed the photo album.
- The user might have created or edited the virtual object and wants to spread it around.
- The user might have received the virtual object from a friend who needs help spreading it around.

In addition to updating the location data and therefore the virtual object's spatial behaviour, the virtual object's time to live might be incremented upon release.

Virtual objects can also be disassociated from a user in response to a negative interaction or non-use by its holder. When a virtual object disassociates from the user, its location will be updated so that it is no longer determined by reference to a user, but rather, the virtual object will be available for other users to collect and associate with themselves. Advantageously, the virtual object may be (or become) visible to all users of the system, providing an opportunity for the virtual object to continue to live on with other users.

Prior to the virtual object automatically disassociating itself from the user, a warning icon could appear to encourage the user to find a good place to drop the virtual object, allowing it to survive.

To incentivise users to help virtual objects propagate, a user can have a "*kiss-of-death*" attribute that tracks the user's propensity to retain objects until they expire (instead of passing them on and thus increasing their lifespan). Using this attribute, where a virtual object decays to its death, the last user that touched it will have his *kiss-of-death* statistic incremented once. When viewing a user's profile, if a user has a high *kiss-of-death* statistic, this indicates that he is very bad at placing virtual objects in good locations and should be less desirable as a host for virtual objects to automatically attach to or those who wish to spread their own viral virtual objects.

After their creation, virtual objects can be edited to include additional or modified modules. Advantageously, the editing of particular virtual objects can be tracked. The system may allow users to take modules from existing virtual objects and reuse these to create new virtual objects. A user might decide to create a new virtual object that "borrows" modules from two or more virtual objects that are associated with her. She can then mix in her own modules if she wants to. This way, logos, pictures, videos, quotes etc. can be recycled and mashed-up to create countless new derivative works.

By tracking where virtual objects or modules within them arise from, a concept like lineage can be implemented. Virtual objects that share a single common ancestor object are said to be of the same *family*. When a virtual object is created, the system tries to link it to a *parent* virtual object. Each virtual object has just one parent, the association being made by the action

of cloning a virtual object. Where a viral virtual object divides, the two resulting virtual objects both have the same parent.

The user can select a virtual object and opt to view all virtual objects that are part of the same family. Accordingly, one virtual object can be used to track potentially thousands of other related virtual objects. They may all be identical, or some may have mutated due to editing. Users can view these relative virtual objects even though they may be very far away, however they cannot edit them remotely.

It will be appreciated that the editing process provides an opportunity for revenue loss, since it is possible that an existing virtual object could be substantially edited in order to avoid paying for a new virtual object. Accordingly, the present system determines the similarity of edited virtual objects compared to their originals, blocking drastic edits which significantly change the original version. For example, an edit changing more than 30% of the virtual object modules may be blocked.

Also a virtual object whose health is less than a predetermined threshold may automatically *merge* with other virtual objects of the same family that are in proximity. This way, the life of the content is extended because one healthy virtual object will last longer than 5 unhealthy virtual objects that are close to death.

In addition to this merging for the purpose of survival, virtual objects may merge for the purpose of aggregating content. This is a separate scenario whereby, for example, a photo montage of one virtual object is automatically aggregated with a photo montage of another, to produce a larger, more interesting photo montage. Virtual objects in the same family may do this when they are in proximity and when some criteria have been satisfied, such as, in the given example, that both virtual objects contain sufficient photos.

The concept of lineage described herein could, in some embodiments, also allows an interaction with one member of a family to affect another member of a family. That is when a virtual object of a family is collected, its children or siblings may be changed in some way: e.g. to change their time to live (longer or shorter) to change their appearance; change their movement behaviour, for instance to cause them to move towards or away from the captured virtual object. A user is generally unable to view a virtual object that is presently associated with another user. However, it is possible to trade virtual objects. Each party is presented with a list of virtual objects that the

counterparty is willing to trade. Each user makes a selection as to the virtual objects that they would like to trade, and based on their respective selections, decide whether the trade should proceed. If the trade proceeds, each virtual object that is the subject of the trade is updated to indicate an association with the respective counterparties. The trading interaction results in the virtual object's association being changed from one user to another. The virtual object's health can also be updated.

To better illustrate some of the variety of ways in which embodiments of the present invention can be implemented, several exemplary uses will be described.

Examples

- 10 As will be appreciated from the foregoing the parameters that are set for a virtual object will define the virtual object's interactions and behaviours. For instance take a virtual object which, in addition to more common data, such as an appearance, location, canvas and module data, owner, health etc, has the following additional parameters defined:

a desired destination or state,

- 15 an action to perform when the destination is reached or the state achieved,

permission only to freely associate only with users belonging to particular user group,
and

- user association parameters that make it become associated with a member of the group that is within its interaction radius from time to time, but to preferentially make associations with
20 users that are moving towards its destination.

This virtual object is intended to make its way to a given location by forming advantageous associations with predefined users.

- Such a virtual object could be used to advertise a sponsor of a sporting team and be "planted" in the home town of the sporting team on the morning of a big game. The virtual object could then
25 attach itself to a member of the sports club and hitchhike to the game with various members along the way. Upon the virtual object's arrival at the event it (possibly only when some critical mass of similar virtual objects have also arrived) can trigger an event, such as the awarding of a

prize to one or more members of the club (e.g. the user that is carrying the virtual object at the time of arrival)

In the context of the invention, each time a member of the club interacts with it, say by picking it up, it's position data is updated to correspond with that user. Thus data defining the virtual

5 object's movement behaviour is modified by virtue of the virtual object's association with a new user. Each handover will also affect the virtual object's lifetime behaviour by increasing its health. Also at each handover the virtual object may zero a timer which defines its maximum residence time with each user, before it starts looking for a new user to jump to. The propensity to jump may also be tied to the number of club members within jumping range.

10 To illustrate how this situation imbues a virtual object with animacy, take the contrasting situation of a virtual object placed at the main train station in a sports club's home town. It is likely that this virtual object will have many club members to associate with and jump between as the supporters make their way to the game. This activity will increase the health of the virtual object over time and it will arrive at the game. In contrast a virtual object for the same team, that
15 is placed at the main train station in the rival's home town will struggle to find club members to carry it, and thus its health will be decremented by the system and it may even die before making the ground. However a small group of club members travelling to the game together from this "enemy territory" could group together to take turns in carrying the virtual object and get the virtual object to the game. If nobody picks it up, the virtual object will expire.

20 A user waiting at a bus stop creates a new virtual object and fixes it at the location of the bus stop. The virtual object is a poem. A subsequent person waiting at the bus stop finds the virtual object, and reads the poem. The original virtual object's health is incremented so it lives on to possibly be read by another user. However, the reader also copies the poem and creates a mobile version of the virtual object. The user wants to credit the original author, and so sets a
25 system rule that every second time their cloned virtual object is read and passed on the original virtual object's health is incremented, so as to increase the original virtual object's chance of surviving. The user takes their new virtual object with them and hands it off to their friend at work. This friend enjoys the poem and passes it to her brother. In response to this interaction the copied virtual object's location data and associated data is updated, and the original virtual
30 object's health is updated.

A teenager wishes to "tag" a wall with their nickname. Whereas the teenager may have traditionally used spray-paint or a similar medium, a degree of notoriety may be achieved by instead creating a virtual object at that location. The virtual object medium may be particularly attractive due to its richness – in particular the ability to embed video and audio at a particular location. The approach potentially also enables classification and filtering of content – for example, "tags" may be visible to users who wish to see them, but filtered for users who do not.

A local band wishes to promote their concert. Instead of sticking a poster to a message board, the band creates a virtual object at the location of the message board. The message includes a sample track recorded live at their last performance, and a wallpaper flyer including the concert date which they can transfer to their mobile device. The virtual object allows the user to easily add the concert date to their calendar. Users can interact with this virtual object at the message board. The band can also leave 10 bubbles at location, or group of locations, say a shop or venue. These virtual objects can be taken away by fans. The bubbles can be set to "go off" or "ring" at a set time to remind the user that is carrying them that the concert is in 7 hours. If a user carries one of these bubbles to the concert, and drops it there the user can be provide with some bonus, e.g. unlocking exclusive content or free entry to the venue itself.

A husband wishes to leave his wife a message for their anniversary. He writes a private note which he anchors at her workplace. The virtual object is accessible to the couple, but not others. When she arrives, she unlocks the virtual object to reveal the message. In response to her unlocking the message the association data of the virtual object is updated to be attached to her. Also a previously hidden, associated virtual object, which renders as a bunch of flowers is updated, so that its rendering behaviour changes, and it appear to spontaneously appear. This is signalled to the woman using the push notification service, so she looks at her user device and sees that she's been given flowers. In fact all users within a radius of the woman could be notified so that the act of "giving" the flowers is seen by the public.

A person wishes to advertise a room in a share house. The person creates a virtual object containing details of the rent, and basic contact details (including a clickable telephone number). The virtual object also contains a password protected section which provides the location of the share house. The virtual object is marked to allow cloning, allowing a person searching for a place to rent to collect contact details without needing to use a pen. Upon contacting the advertiser, the prospective flatmate is provided with a password enabling them to easily locate

the house. The map interface of the client application can be user to direct the user to the house for a meeting with her potential housemates.

It will be appreciated that the approach opens up new possibilities for collaboration by groups of people who are only very loosely associated, for example, associated by visiting a particular geographic location. This allows a large number of users to collaborate to create various types of virtual objects, for example, a picture wall containing pictures contributed by each of the users.

This loose association facilitates uses that require a degree of anonymity. For example, a user may create a virtual object containing an anonymous confession.

It is important to appreciate that virtual objects are not merely characterised by their geographic location, but rather can be characterised by their associated user. This distinction opens up further possible uses. One such use is the chain or forwarded virtual object. This type of virtual object is created with forwarding instructions of some type (for example, forward to ten of your friends, forward to Barack Obama). These virtual objects rely primarily on social, rather than geographical associations in order to propagate between users.

The virtual objects maybe created for viewing by a future generation, such as a grandchild, or only become visible after a fixed period of time, in the nature of a time capsule.

Each time the virtual object is passed on the virtual object's user association is updated. In the event the virtual object has a target user assigned to it, the virtual object behaviour can be set to change dramatically when that person becomes associated with the user, for instance content that was previously locked can be unlocked, or it could spawn other virtual objects as a result.

As noted above, the present system provides commercial opportunities for businesses to either promote their real world goods and services or possible distribute services, virtual objects, e.g. containing media, via the artificial reality system. By tracking virtual object use and dissemination the system operator can charge for this service. The predominant mechanism envisaged for this is to apply charges for providing a virtual object with "life". This can be achieved by charging the owner of a commercial virtual object each time a virtual object's health parameter is incremented. When a virtual object is created, it will have some pre-set initial health

parameter. The creator can be charged for the amount of health given to the virtual object. Because the health of the virtual objects is decremented over time, charging by the health parameter cause an advertiser to have to pay, so long as they want their advertising virtual object to remain in the augmented reality space. Incrementing health upwards upon certain user
5 interactions also allows the system operator to charge an advertiser when a virtual object attracts a user's attention, thus tying the cost of advertising to its success.

As will be understood by those skilled in the art, aspects of the present invention can be implemented on a wide range of fixed or mobile computing hardware. That hardware can run any one of a number of suitable operating systems. Moreover aspects of the invention
10 implemented in software could be written in any suitable programming language. Accordingly the present invention should not be considered to be limited to the hardware, operating system or software implementations, which are described herein as examples.

It will be understood that the invention disclosed and defined in this specification extends to all alternative combinations of two or more of the individual features mentioned or evident
15 from the text or drawings. All of these different combinations constitute various alternative aspects of the invention.

CLAIMS

1. A method in a computing system, the computing system including:

a data storage system storing data representing at least one virtual object, said data

5 representing the at least one virtual object including virtual object behaviour data and virtual object location data, and

a data processing system configured to process data stored in the system, and to communicate with a user device via at least one communications channel; said method including:

10 identifying an interaction between a virtual object and a user; and

updating virtual object behaviour data relating to one or more virtual objects in response to the identified interaction.

2 The method of claim 1, wherein the step of identifying an interaction comprises a step of receiving data identifying a virtual object.

15 3. The method of any one of the preceding claims, wherein the step of identifying an interaction comprises a step of receiving data representing a user input signifying a user initiated interaction with a virtual object.

4. The method of claim 3, wherein the user initiated interaction includes any one of:

the user using the virtual object;

20 the user copying the virtual object;

the user retaining the virtual object;

the user viewing the virtual object;

the user accessing the virtual object;

the user accepting the virtual object;

the user capturing the virtual object;

the user moving the virtual object;

the user modifying the virtual object; and

the user releasing the virtual object.

5 5. The method of claim 3, wherein in response to receiving the user initiated interaction, the computing system performs a step of transmitting data including data enabling or facilitating the rendering or other reproduction of the virtual object.

6. The method of any one of the preceding claims wherein the step of identifying an interaction comprises:

10 receiving data representing a location of a user;

determining a proximity between a user and a location corresponding to the virtual object location data; and

in the event that the proximity is within an predetermined radius of the virtual object and/or user, determining that an interaction has occurred.

15 7. The method of any one of claims 1 to 6, wherein the step of updating virtual object behaviour data relating to one or more virtual objects, includes updating association data relating the virtual object to the user.

8. The method of claim 7, wherein the step of updating association data relating the virtual object to the user, includes updating the virtual object location data for the virtual object.

20 9. The method of any one of the preceding claims further includes checking interaction permission data of at least one of the virtual object or user.

10. The method of claim 3, wherein in response to receiving the user initiated interaction, the computer system performs a step of copying the virtual object.

25 11. The method of either of claims 7 or 8, wherein the step of updating the association data comprises associating the user with the virtual object.

12. The method of either of claims 7 or 8, wherein the step of updating the association data comprises dissociating the user from the virtual object.

13. The method of any one of the preceding claims, wherein updating the virtual object behaviour data includes modifying the data representing the virtual object such that its
5 location data is determined by reference to a user rather than a location.

14. The method of claim 1 wherein updating the virtual object behaviour data includes modifying the data representing the virtual object such that its location data is determined by reference to a location rather than a user.

15. The method of any one of the preceding claims, wherein modifying the virtual
10 object behaviour data includes a step of modifying data corresponding to the rendering or other reproduction behaviour of the virtual object.

16. The method of any one of the preceding claims, wherein modifying the virtual object behaviour data includes a step of modifying data corresponding to the interaction behaviour of the virtual object.

15 17. The method of any one of the preceding claims, wherein modifying the virtual object behaviour data includes a step of modifying data corresponding to the health of the virtual object.

18. The method of any one of the preceding claims, wherein modifying the virtual object behaviour data includes a step of modifying data corresponding to a virtual object related
20 to the virtual object.

19. The method of claim 16 wherein the step of modifying the interaction behaviour data includes a step of updating or determining user permission to interact with the virtual object.

20. The method of claim 16, wherein the step of modifying the interaction behaviour data includes a step of updating or providing an interaction type specifying the nature of
25 interaction between a user and virtual object; or between virtual objects.

21. The method of claim 16, wherein the step of modifying the interaction behaviour data includes a step of updating an interaction radius applicable to the virtual object.

22. The method of claim 15 wherein the step of modifying data corresponding to the rendering behaviour or reproduction behaviour includes providing data representing a style of rendering the virtual object.

23. The method of claim 15 wherein the step of modifying data corresponding to the rendering behaviour includes providing data representing a visibility radius applicable to the rendered virtual object.

24. The method of claim 17 wherein a step of modifying data corresponding to the health of the virtual object includes updating data corresponding to a time to live of the virtual object.

25. The method of any one of the preceding claims wherein a step of updating virtual object behaviour data includes updating data that indirectly modifies a behaviour of the virtual object.

26. A method as claimed in claim 24 which includes incrementing, decrementing or re-setting a time to live of a virtual object in response to a user interaction with a virtual object.

27. A method as claimed in any one of the preceding claims wherein the data storage system stores data representing a health parameter of a virtual object; and the method includes:
updating the health parameter of one or more virtual objects from time to time.

28. The method of claim 27 where the health parameter of one or more virtual objects are updated periodically.

29. The method of claim 28 wherein the step of updating includes decrementing the health of one or more virtual objects.

30. The method of claim 27 in which the health parameter of substantially all virtual objects are decremented periodically.

31. The method of any one of the preceding claims wherein the step of updating virtual object behaviour data relating to one or more virtual objects includes updating data relating to at least one virtual object associated with the virtual object involved in the identified interaction.

32. The method of claim 29 wherein the virtual object(s) to be updated are associated with the virtual object by any one or more of the following attributes:

- virtual object creator;
- virtual object antecedent or lineage;
- 5 • virtual object owner;
- virtual object proximity;
- virtual object type; and
- a user/creator/owner defined association.

33. An augmented reality system including:

10 a data storage system storing data representing:

- at least one virtual object, including virtual object behaviour data and virtual object location data;
- at least one user; and
- 15 • a data processing system configured to implement a method as claimed in any one of claims 1 to 32.

34. The augmented reality system of claim 33 wherein the data storage system includes media storage for storing media virtual objects associated with one or more virtual objects.

35. The augmented reality system of claim 33 which further includes at least one
20 interface to a communications network to receive and/or transmit data from and/or to a user client device.

36. The augmented reality system of claim 35, wherein said interfaces are adapted to exchange any one or more of:

interaction data, location data, virtual object data with the user device.

37. A client for an augmented reality system, the client providing an augmented reality interface including:

an user interface portion corresponding to a real world scene and a portion representing
5 one or more virtual objects;

a user input portion enabling a user to interact with a virtual object according to behaviour data corresponding with a virtual object; and

a communication portion confined to communicate with an augmented reality system to exchange at least interaction data and behaviour data.

10 38. The client of claim 37 which is configure to perform any one or more of the following interactions:

the user using the virtual object;

the user copying the virtual object;

the user retaining the virtual object;

15 the user viewing the virtual object;

the user accessing the virtual object;

the user accepting the virtual object;

the user capturing the virtual object;

the user moving the virtual object;

20 the user modifying the virtual object; and

the user releasing the virtual object.

39. The client of either of claims 37 or 38 which is adapted to interact with a system of any one of claims 33 to 36.

40. The client of any one of claims 37 to 39 which is implemented in a mobile computing device.

41. The client of claim 40 wherein the mobile computing device includes a locator component configured to determine the location of the mobile computing device.

5 42. A client of any one of claims 37 to 41 comprising a software application configured to be run on a computing device.

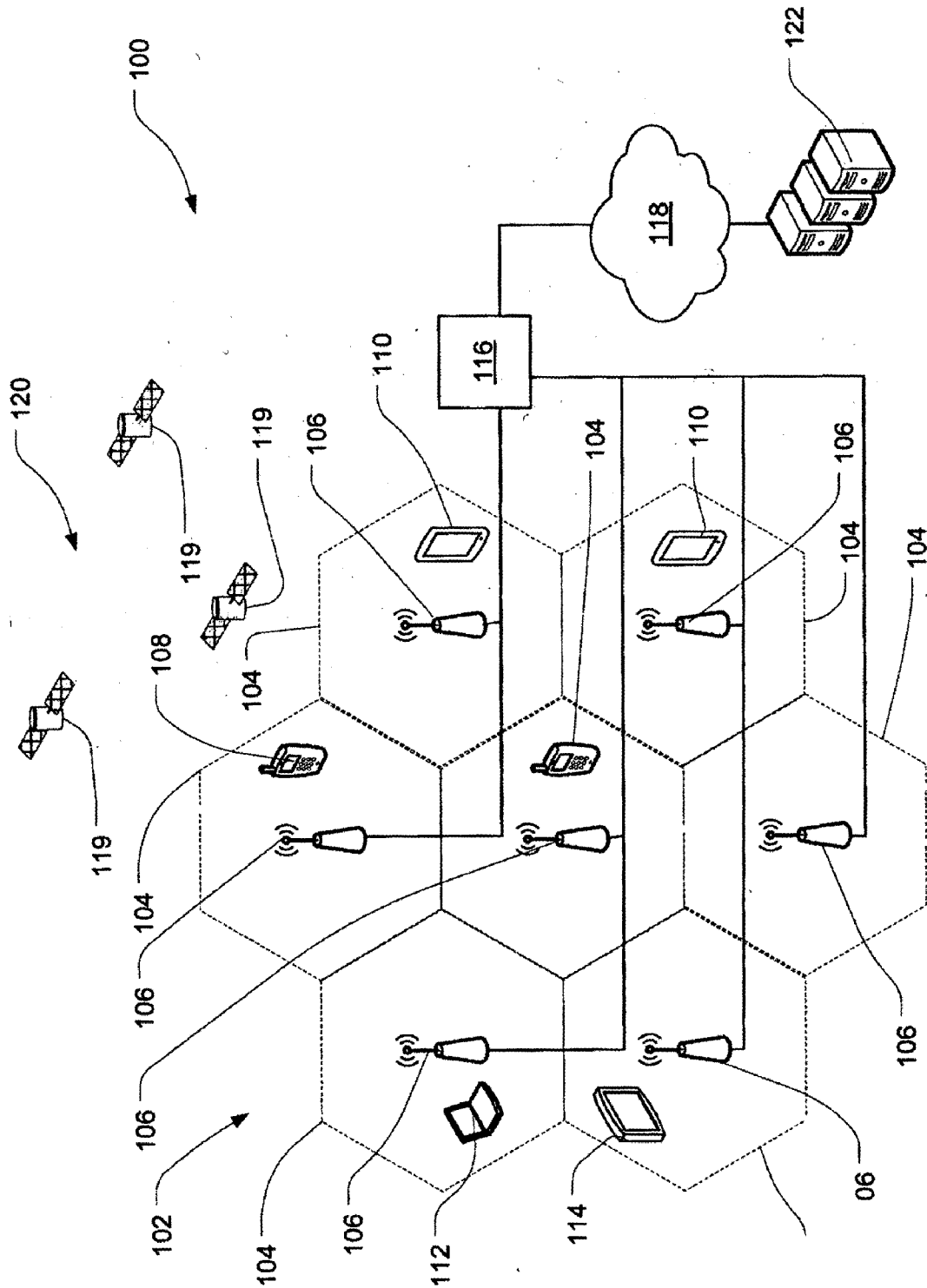


FIG. 1

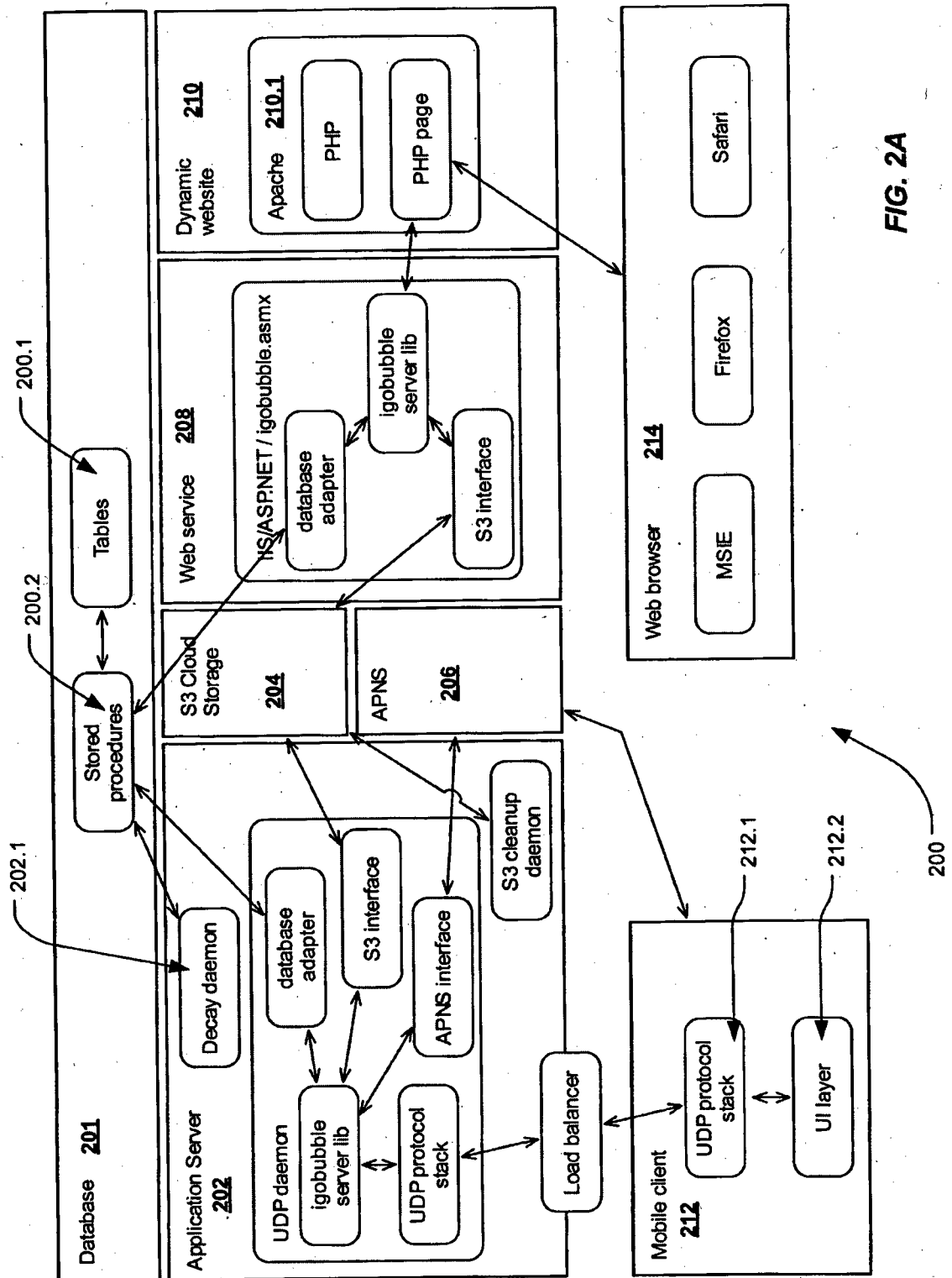


FIG. 2A

3/9

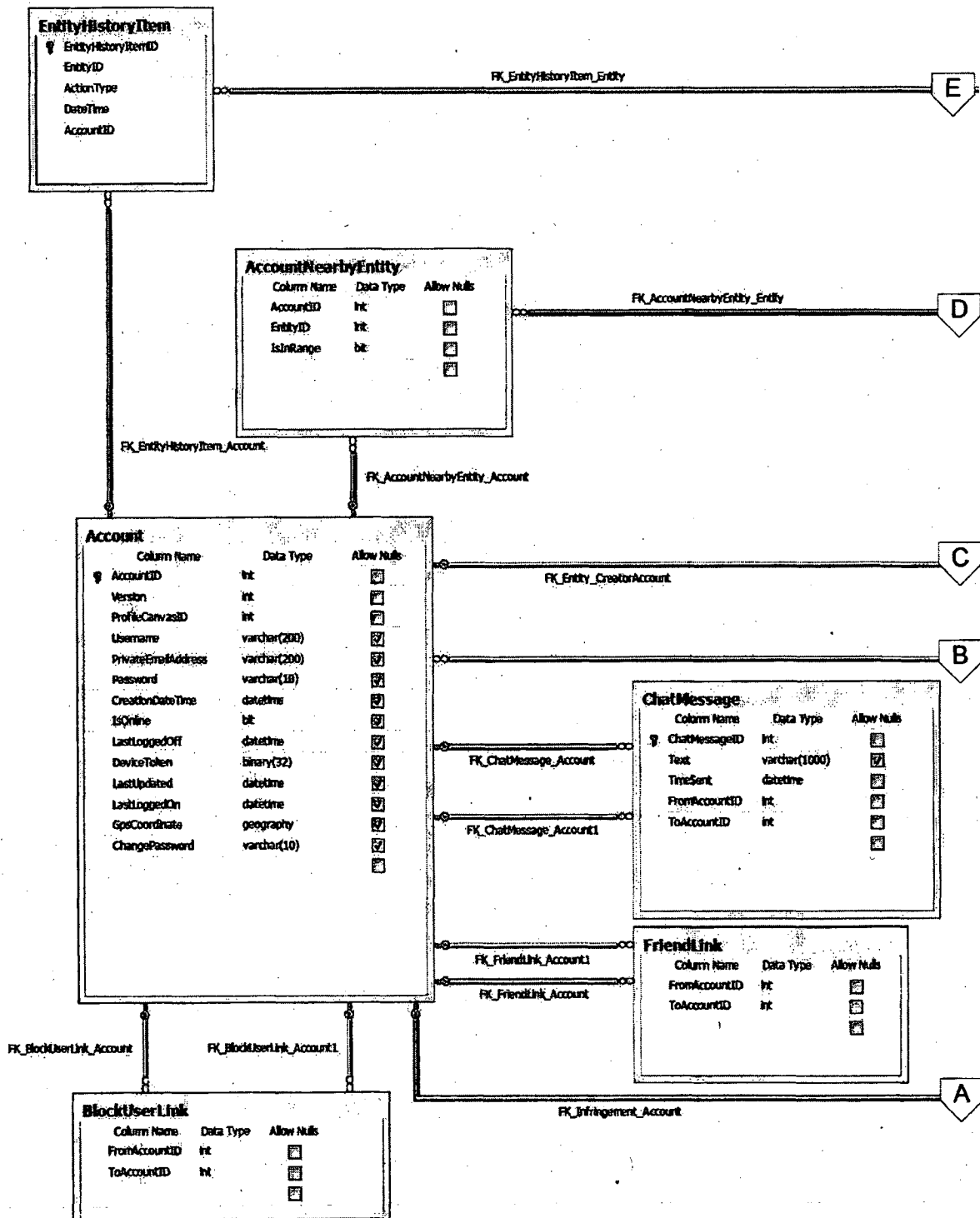


FIG. 2B

4/9

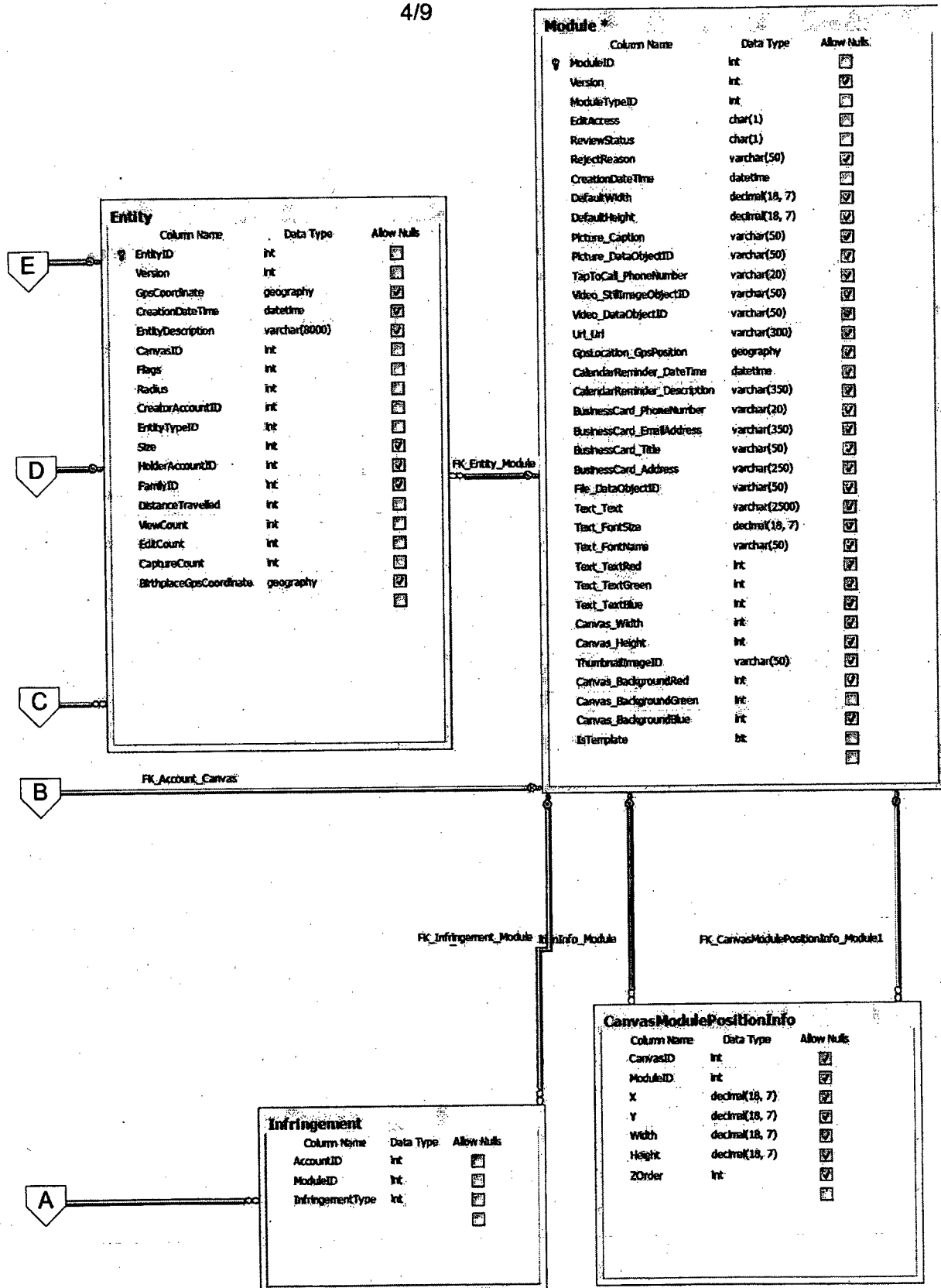


FIG. 2C

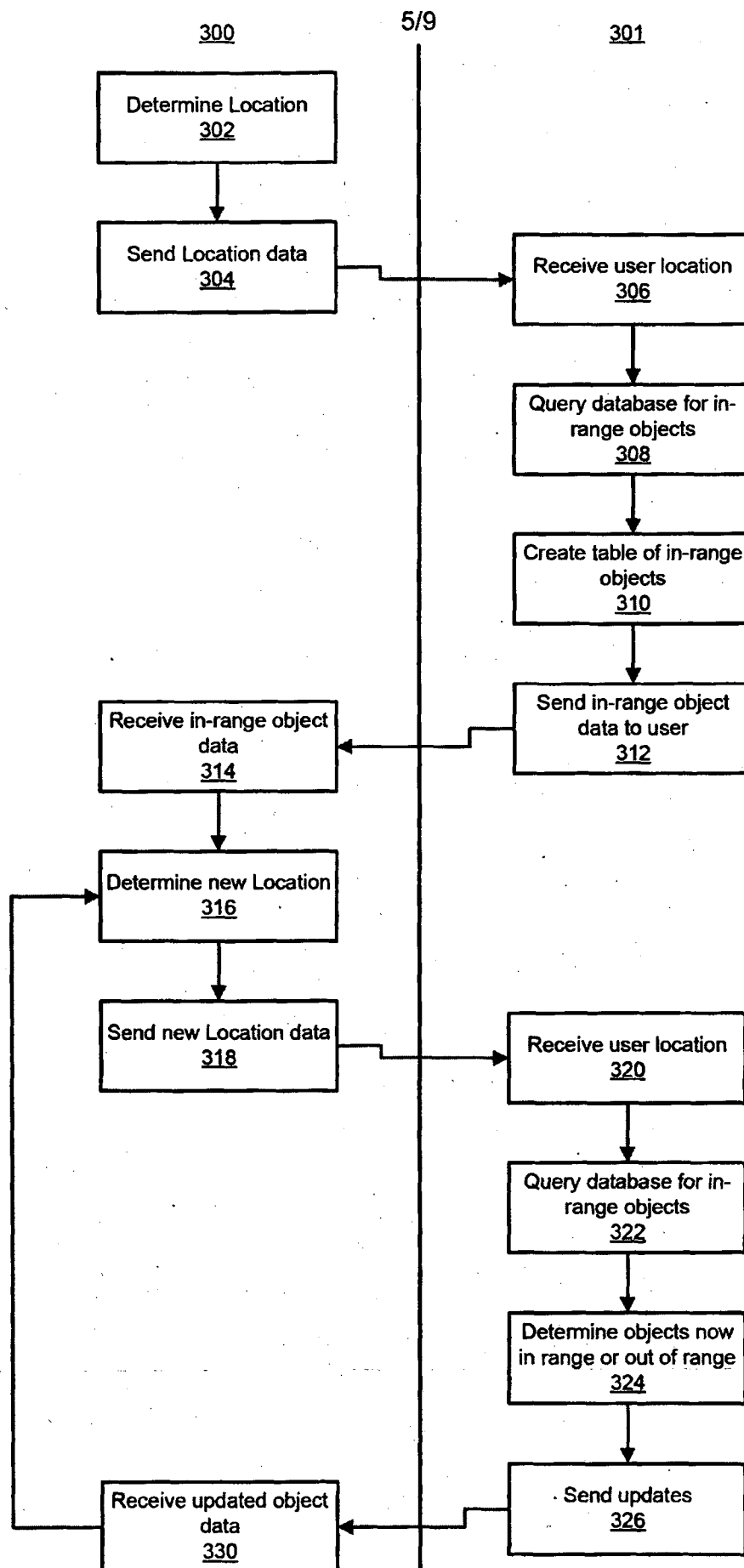


FIG. 3A

6/9

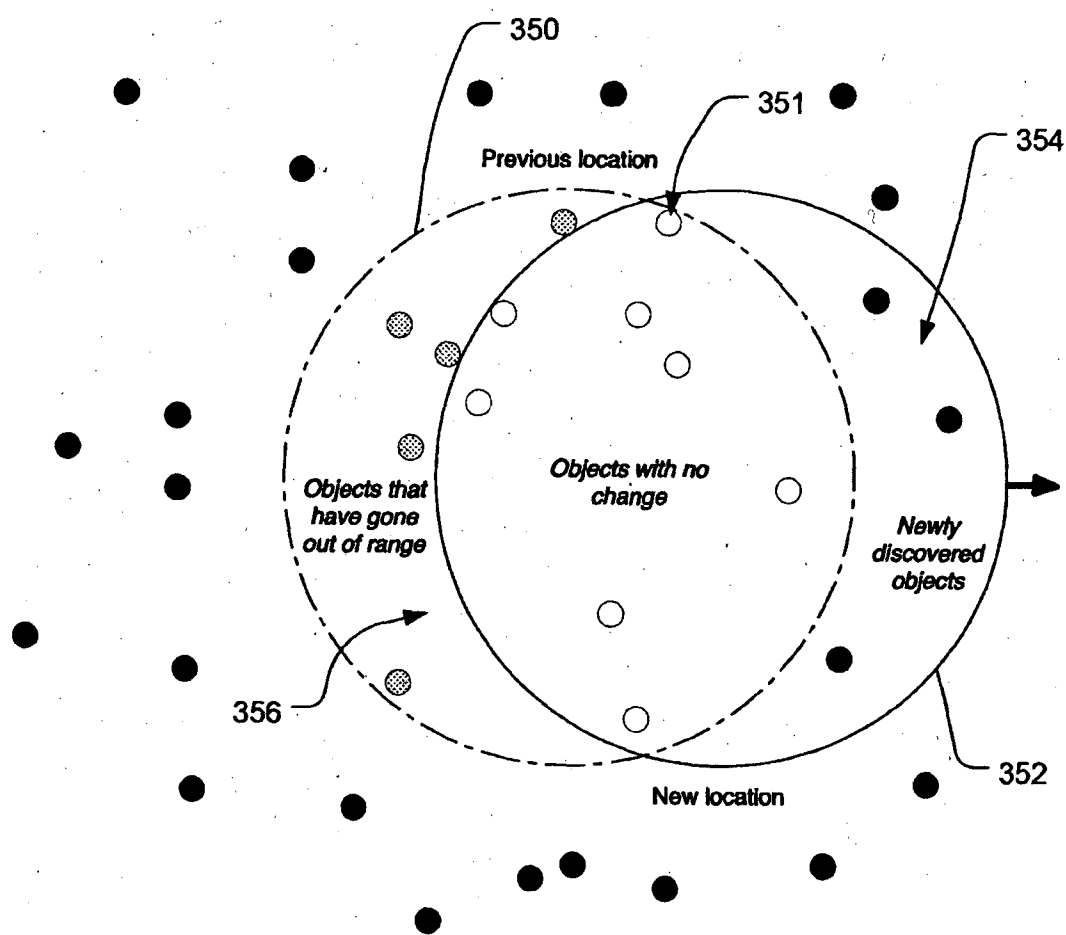


FIG. 3B

7/9

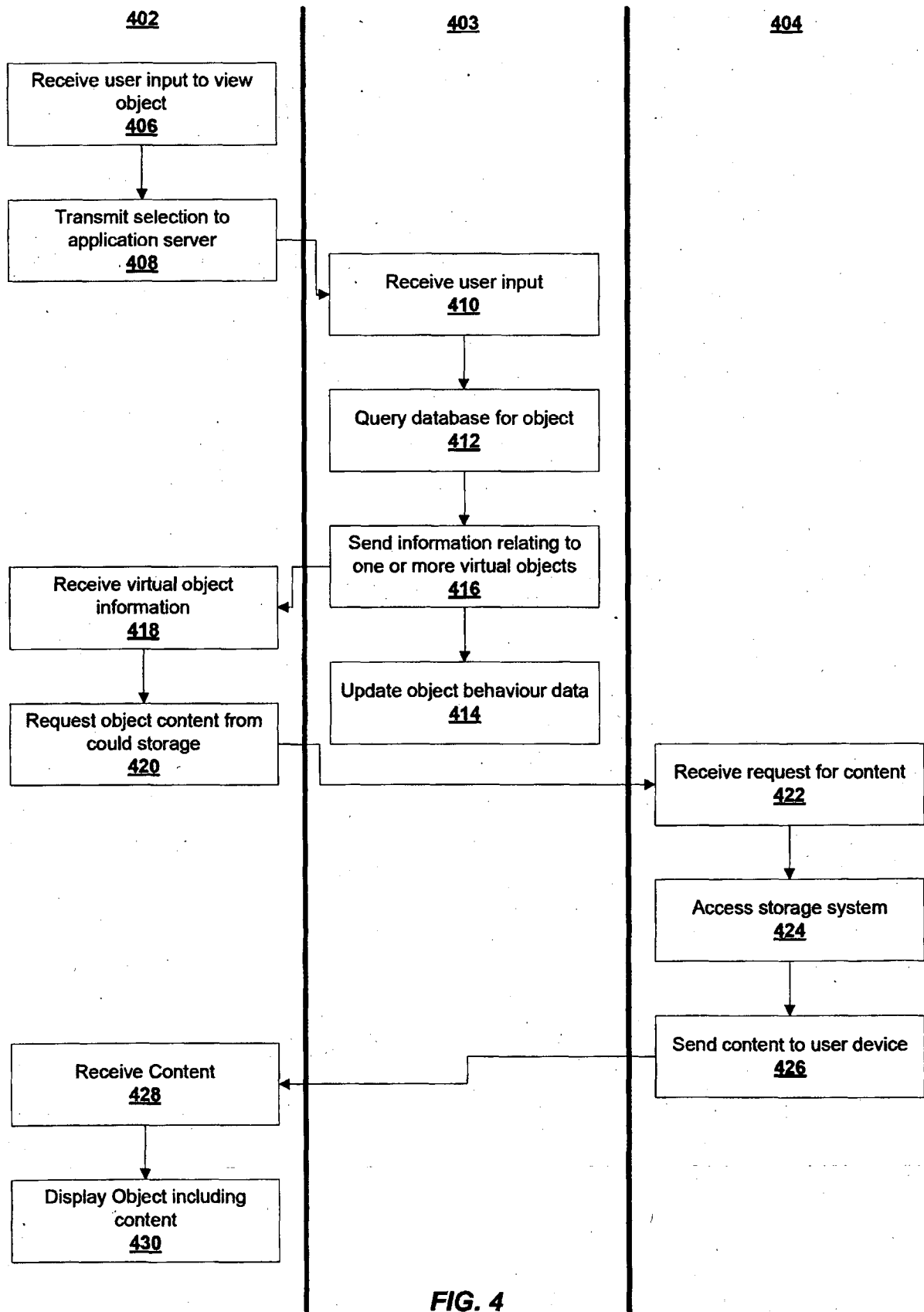
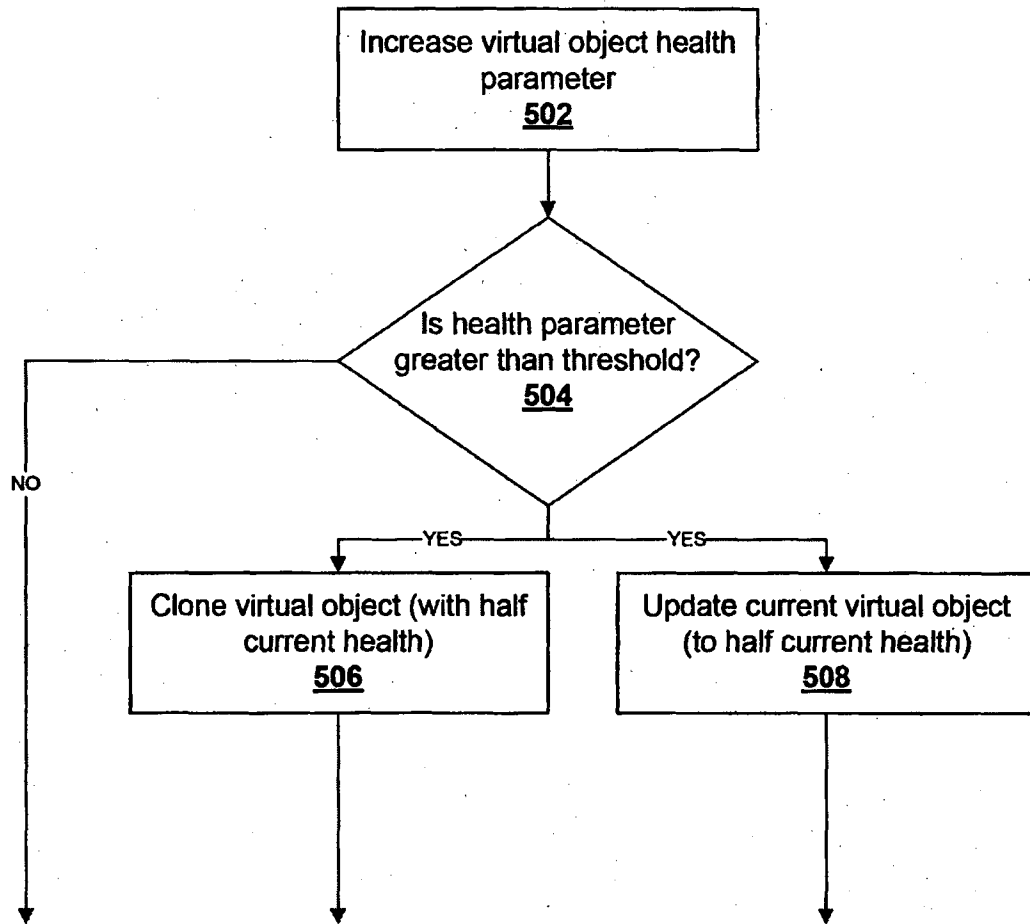


FIG. 4

8/9

**FIG. 5**

9/9

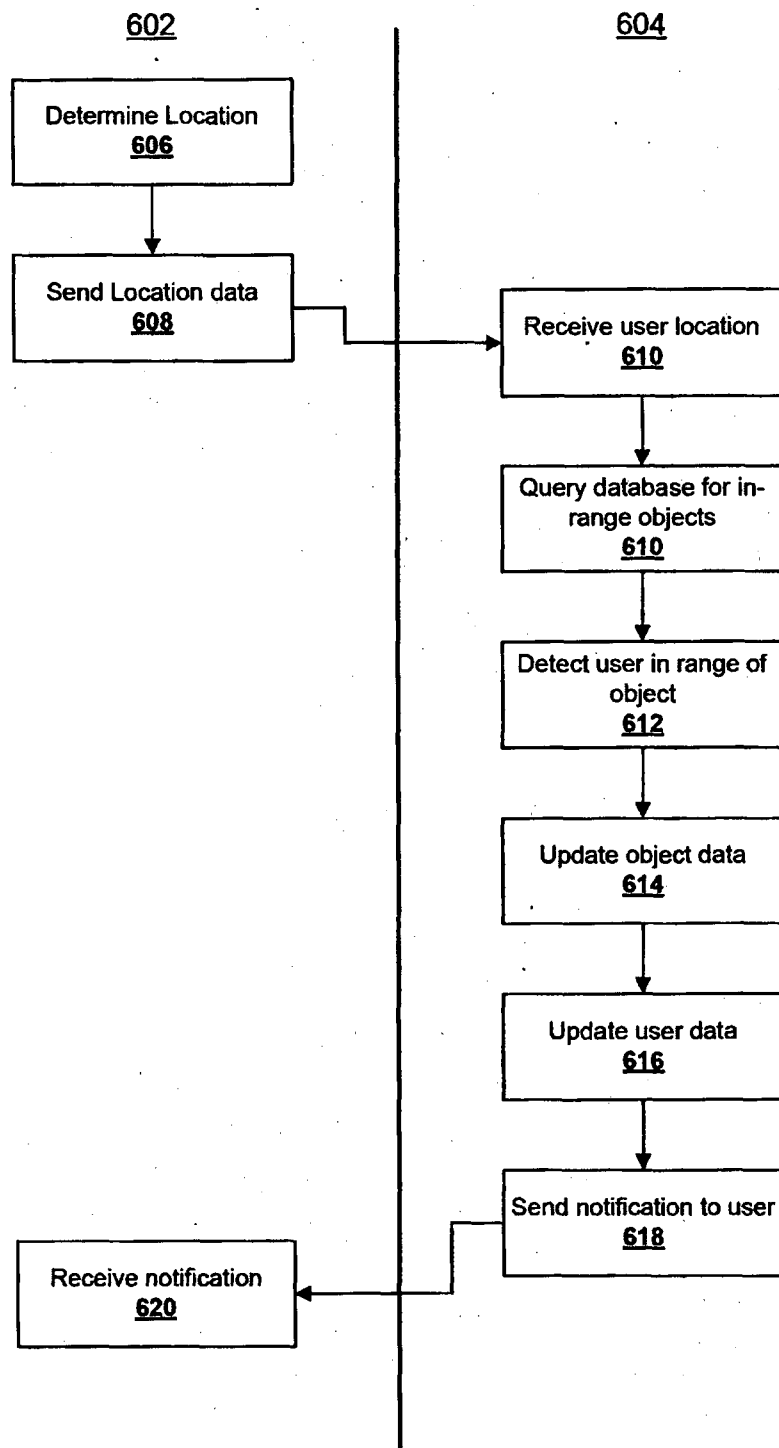


FIG. 6

INTERNATIONAL SEARCH REPORT

International application No.

PCT/AU2012/000495

A. CLASSIFICATION OF SUBJECT MATTER G06T 19/00 (JAN 2011) G06F 3/048 (OCT 2005)		
According to International Patent Classification (IPC) or to both national classification and IPC		
B. FIELDS SEARCHED		
Minimum documentation searched (classification system followed by classification symbols)		
Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched		
Electronic data base consulted during the international search (name of data base and, where practicable, search terms used) WPI, Google, Patent Lens; Keywords: (augment, virtual, reality, world, scene, object, avatar, user, character, entity, update, adapt, change, modify, evolve, mutate, behaviour, response, action, movement, interact, event, stimuli, contact, data, information, proximity, location, position, radius, health, life, age, relation, associate, link, copy, clone, duplicate, replicate, reproduction, render, permission, parameter, definition, property, attribute, state) + similar keywords		
C. DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	Documents are listed in the continuation of Box C	
☒ Further documents are listed in the continuation of Box C ☐ See patent family annex		
* Special categories of cited documents: "A" document defining the general state of the art which is not considered to be of particular relevance "E" earlier application or patent but published on or after the international filing date "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified) "O" document referring to an oral disclosure, use, exhibition or other means "P" document published prior to the international filing date but later than the priority date claimed	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art "&" document member of the same patent family	
Date of the actual completion of the international search 05 July 2012		Date of mailing of the international search report 06 July 2012
Name and mailing address of the ISA/AU AUSTRALIAN PATENT OFFICE PO BOX 200, WODEN ACT 2606, AUSTRALIA Email address: pct@ipaustalia.gov.au Facsimile No.: +61 2 6283 7999		Authorized officer Olutope Omogbenigun AUSTRALIAN PATENT OFFICE (ISO 9001 Quality Certified Service) Telephone No. 0262832876

INTERNATIONAL SEARCH REPORT		International application No.
C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		PCT/AU2012/000495
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 7685518 B2 (MATSUDA et al.) 23 March 2010 See whole document, especially abstract, col. 3 lines 48-52, col. 6 line 59 – col. 7 line 15,31-42, col. 9 lines 3-9, col. 10 lines 8-33,41 – col. 11 line 9, col. 13 lines 1-4,24-30, col. 15 lines 1-6,53-67, col. 18 lines 5-7,27-29, fig. 8	1-17,19-30
Y	abstract, col. 3 lines 48-52, col. 6 line 65 – col. 7 line 15,31-42, col. 9 lines 3-9, col. 10 lines 8-12,19-26,52-57, col. 13 lines 24-30, fig. 8	18,31,32,33-36,39
X	US 2010/0130296 A1 (ACKLEY et al.) 27 May 2010 paragraphs [0026],[0032],[0036],[0040],[0042],[0043], fig. 1	37,38,40,42
Y	paragraphs [0026],[0032],[0036],[0040],[0042],[0043], fig. 1	33-36,39
X	WO 2008/130842 A1 (UTBK, INC.) 30 October 2008 See whole document, especially abstract, paragraphs [0056],[0065],[0089],[00114]-[00120],[00452]-[00454],[00497]-[00501],[00564]-[00568], claim 13	37,38,40-42
Y	paragraphs [0077],[00171],[00275],[00483],[00652],[00653]	33-36,39
Y	US 7409647 B2 (ELBER et al.) 05 August 2008 See whole document, especially col. 14 lines 25-29, col. 15 lines 9-14, claims 23-25, fig. 11	18,31,32
A	US 2010/0070859 A1 (SHUSTER et al.) 18 March 2010 See whole document, especially paragraphs [0014],[0021],[0024]	9,10,19
A	US 2010/0304862 A1 (COLEMAN et al.) 02 December 2010 See whole document, especially paragraphs [0022],[0029]	28-30