Abstract: An information management system comprises an information database (5) operable to store data records (82); an interface component (3) operable to generate user interfaces enabling a user to input data and access and manipulate data records (82) in said information database (5); a context data store operable to store data identifying a user and the operations performed by a user; and a configuration store (1, 2) operable to store configuration data identifying specific operations to be performed in response to user input when identified data is stored in said context data store; wherein said interface component (3) is responsive to user input to generate user interfaces enabling a user to input data and access and manipulate data records (82) in said information database (5) on the basis of said user input, context data stored in said context data store and configuration data stored in said configuration store.

Title: INFORMATION MANAGEMENT SYSTEMS

Fig 1

1. Configuration Component
2. Specialist Screens Component
3. Interface Component: Contains the various Interfaces which permit the normal user to interact with the User Data
4. Deployment Licence
5. System & User Data Component(s)
6. Admin User Accessible Logic and Interface Component
Information Management Systems

This invention relates to computer-based information management systems, particularly to the structure of such systems such as to facilitate specifying and configuring them to meet users' requirements.

Small and medium scale organisations ("SMO's") in the private, public and voluntary sectors represent a huge worldwide market for both bespoke and vertical-market based Integrated Information Software Systems ("IISS's"). Organisations that do not deploy suitable and appropriate IISS's in a high wage cost economy such as the UK often hit a "glass ceiling". This glass ceiling is caused in part by the difficulty in managing complexity when the organisation grows beyond a certain point and in part by increasing difficulty in maintaining efficient risk management and quality control. In order to grow beyond a certain point, SMO's need to deploy an IISS.

Appropriate, easy-to-use and price-competitive IISS's will find a ready market. However, the potential is being constrained by the complexity of the sales process, the uncertainty of implementation and the fact that the cost of developing, marketing and deploying well-designed systems is high. Each system is generally designed from scratch for each vertical market. Software developers must write computer code in an appropriate language to give the required functionality, logic and appearance. Once written such systems are difficult to adapt without re-coding. Although such systems will often include a limited degree of configurability, the elements which can be configured and ranges over which they can be configured generally need to be decided in advance. In other words configuration is generally constrained to be within a pre-defined closed system.

In short, current methods of specifying, configuring, developing and deploying IISS's, where they exist, are slow, uncertain, inefficient and expensive.

Most SMO's are now computerised with a mixture of word processing, accounts packages, email, web browser and spreadsheet applications. Some have specialist systems, often free standing and non-integrated. Increasingly the need to "join up" their systems or migrate into an IISS is recognised. In such an IISS, all key information is held in a central system,
able, if necessary, to link into and communicate with subsidiary local information systems and similar systems within other organisations. Such IISS's, where deployed, are rarely easy to use, are generally expensive and often provide an unsatisfactory user/purchaser experience, most likely for the reasons stated above.

Deployment of an IISS offering a fulfilling, empowering and appropriate user experience at a competitive price would give substantial competitive and economic advantages to that SMO.

Any IISS may, with use of the appropriate technology, be deployed over a local area network, a wide area network, or via any other means of connecting computers. It may employ web browsers or other means for displaying interfaces.

In one embodiment the present invention may provide an information management system structure which overcomes or at least mitigates the problems set out above.

When viewed from a first aspect the invention may provide an information management system adapted to generate a plurality of user interfaces from a common interface structure, said system comprising means for retrieving data essential for the generation of each of said interfaces from a separate interface configuration data storage means, the system further comprising user operable means for modifying the data stored in said interface configuration data storage means.

When viewed from a second aspect the invention may provide an information management system comprising a plurality of user interface instances which differ from each other under the control of reconfigurable data such that any interface can be transformed into any other by altering said reconfigurable data.

The invention may extend to a method of operating a computer system comprising generating a plurality of user interfaces from a common interface structure wherein the step of retrieving data essential for generating said user interfaces from a separate interface configuration data storage means.
Thus it will be seen by those skilled in the art that in accordance with the invention the individual user interfaces, e.g. individual screens or windows, with which a user can control and receive information from the information management system are generated based on user-defined data. This allows a single master interface structure to be used to generate all of the user interfaces simply by retrieving the appropriate data from the interface configuration data storage means. Such a structure, in which a user interface is generated by retrieving data stored by a user in order to build the interface, has several advantages over conventional arrangements in which user interfaces are hard-coded - i.e. generated from object code. For example it allows a robust basic master user interface structure to be developed which is retained as it is implemented for each particular instance of the user interface. It also allows the rapid creation of a large number of interfaces which may be quite different from one another in detailed appearance and functionality but which retain a common look and feel, without coding them separately.

Most significantly systems in accordance with the invention allow a user to create a series of user interfaces without requiring that user to have the skills required to code the interface from scratch. This opens up the possibility of a user developing an integrated information management system without requiring detailed computer programming knowledge. Clearly in this context the term "user" is intended in a broader sense to encompass a developer developing a system rather than just an end user of the system. Thus herein the term "user" is intended to include a developer developing a particular information management system by entering data into the configuration data storage means referred to above, as well as an end user.

A further advantage achievable in accordance with the invention is that any given user interface can easily be altered in appearance and/or functionality in dependence on a range of conditions e.g. navigation history, current data module or even user inputs. This is preferably effected by changing the appropriate data in the interface configuration data storage means or by means of calculations which change the appropriate data output by the interface configuration data storage means.

In any given practical implementation of the system a plurality of different common interface structures might be provided, e.g. for different display resolutions or sizes.
However such embodiments still retain the advantages of being able to provide a high quality user interface with a consistent appearance and behaviour but with minimum development time and effort being required.

Preferred embodiments of the invention provide a method of using computer software to implement a business process which permits and enables rapid specification, development, configuration and deployment (together "Development") of easy-to-use IISS’s for a wide range of different customer organisations.

In accordance with the invention user operable means are provided for entering or manipulating the data stored in the interface configuration data storage means. In preferred embodiments this takes the form of a Rapid Development and Deployment Environment ("RDDE") designed to enable the system development method of the invention to function efficiently. Preferably the RDDE is arranged to manipulate other configuration data.

In preferred embodiments of the invention the information management system comprises one or more modules constituting a core system, including as a minimum the common interface structure referred to above, wherein the core system is arranged to be configured by retrieving data from data storage, e.g. the interface configuration data storage means. The RDDE thus provides a way to configure and change the core System so that every customer can be supplied with a version of the same core System, configured to their own individual requirements by configuration data. The ability to configure a system which is achievable in accordance with the invention means that widely differing systems may be based on a single robust core system and can be implemented by manipulation of user data which does not jeopardise the functioning of the core components.

Furthermore every customer can receive easy-to-install updates of the core system without affecting their configuration data and so without impacting on the appearance and functionality of their system; and therefore without jeopardising the stability or reliability of their system. Accordingly every customer can benefit from the improvements developed for other customers merely by installing an update.
Both the RDDE and the core system use a significant level of indirection - the ability to reference something using a name, reference, or container instead of the value itself - to achieve flexibility and configurability. In consequence, preferred embodiments of the invention allow any SMO to be supplied with an IISS that exactly meets its agreed needs in a more timely and cost effective manner than is currently the case.

Further, once a developer has sufficient knowledge of the requirements of a particular Vertical Market Sector, for example "British Not For Profit" or "French Engineering Design", the process of developing and releasing a mass market product expressly meeting the needs of that Vertical Market Sector will be relatively quick and cheap, again providing a significant commercial advantage for such a product. This also offers the significant advantage to customers in that their system can be regularly upgraded with later versions of the core modules which could be under continuous development for other customers, even though the system for which a particular module was developed may be very different (as a result of the configuration achieved through the configuration data).

Preferably means are provided for editing the common interface structure. In some preferred such embodiments this editing comprises the ability to remove one or more elements from the common interface structure. For example in a given implementation it might be desirable to be able to remove one or more entirely inactive elements from all interfaces which can be achieved by removing said one or more elements from the common interface structure. In some preferred embodiments the means for editing the common interface structure allows a user to adjust the position of one or more elements. Again this would be a global change affecting all instances of the interface.

The user interfaces will typically comprise one or more interaction objects, e.g. buttons, for invoking a particular associated action. Preferably the appearance of said interaction objects and the action associated with them are determined by the data stored separately in a button configuration data storage means, which is editable by user operable editing means. This allows great flexibility to a development user in the functionality which can be provided, without requiring access to or knowledge of the underlying code. It also allows the operation of the interaction objects easily to be altered during running of the
system by simply allowing the system to alter the data stored in the button configuration and function library data storage means.

A further advantage afforded by the set-up outlined above is the ease with which text or sounds associated with the interaction objects can be changed en masse, e.g. for an alternative language module. No access to the underlying code is required, it is sufficient simply to replace the data at predetermined points in the button configuration data storage means with the new data; or if the data is already stored (i.e. the system is loaded with another language module), simply to re-address it.

In accordance with the invention the user interfaces - e.g. screens or windows - with which an end user can interact with a information management system can be constructed using data in the interface configuration data storage means. In preferred embodiments a similar arrangement of data storage is employed to navigate between windows, i.e. to implement the topographical layout of the information management system. Accordingly preferably the information management system further comprises a routing configuration data storage means and user operable means for modifying data stored in said routing configuration data storage means, the system being adapted to generate a second user interface after a first user interface in dependence on the data stored in said second configuration data storage means.

Such embodiments of the invention are advantageous since they allow a high degree of flexibility in the layout and order in which screens etc. are accessed without any permanent links being written into the underlying software. This makes the system robust to changes - e.g. the insertion or removal of an additional screen. It also allows for example different users, e.g. those having different security levels, to access screens or not access screens in different orders or to skip some. It also reduces the risk that an unauthorised user can accidentally access a screen which they are not supposed to. This is a frequent hazard in conventional information management systems in which the navigation between screens is embedded in hard-coded logic which can be very hard to verify fully in all its permutations. By contrast in accordance with the aforementioned preferred embodiments of the present invention, the navigation logic can simply be
realised with a matrix of connections, replicated for each user or class of user with appropriate modifications.

Such arrangements are believed novel and inventive in their own right and thus when viewed from a further aspect the invention provides an information management system adapted to generate a first user interface including a plurality of interaction objects, each of said interaction objects being associated with a data table stored in a configuration data storage means, at least one of said interaction objects being configured to generate a further user interface in dependence on the data stored in said data table, the system further comprising user operable means for modifying the data stored in said configuration data storage means.

Preferably the system comprises window configuration data storage means for storing data used to determine the appearance and characteristics of user interface or display windows together with user-operable means for manipulating the configuration data in said storage means. For example the size or location of a window or whether it is printable and if so in what format may be stored in the storage means - e.g. a data table. By storing the data separately from the code used to generate the window, high configurability along with high reliability is achieved. In some preferred embodiments some of the window data may be altered indirectly by an end-user — e.g. by recording the size and position of a window when the end-user closes it so that the size and position of the window is restored when the window is next opened.

Preferably the system comprises means for storing relationship data determining which other records a newly created record is automatically linked to and user-operable means for viewing and editing said relationship data. This allows the automatic creation of records having the correct desired relationships with other user data records within the IISS in a potentially unlimited number of different circumstances.

Preferably the system is arranged such that at least some and preferably all text appearing on any user interface is stored in a separate textual data storage means and is retrieved when the associated user interface is generated. This makes it easy to provide a plurality of versions of each word or text string in different languages.
Preferably the system comprises means for storing standard fields identification data and user-operable means for manipulating the same. This allows users to use standard fields, suitably relabelled where appropriate, to store end-user data. This has the advantage that the underlying core system can be made to function robustly without being affected by configuration and customisation for users. It supports the underlying philosophy of configuration and customisation of the system rather than tying the system to a rigid database structure.

Preferably the system comprises means for storing column sort data separate from the data itself. The column data might include the order in which the data is to be sorted and the labels associated with the column. This feature makes it easy to provide new data columns in a user interface since essentially the functionality is contained in the column sort data storage means.

The system could be implemented in a number of architectures such as on a standalone data processor, a local or wide area network or a remote server delivered via the internet. Where implemented over a network this could be for example using a client-server model, a peer to peer network, or any other configuration. In presently preferred embodiments the system is implemented over a client-server network wherein the interface configuration data storage means is/are provided on a client workstation or on the server but in the latter case mirrored on the client workstation. This reduces network traffic and thus improves performance. Preferably some or all of the other configuration data storage means referred to hereinabove is provided or mirrored on a client workstation. Any individual configuration data storage means or sub-combination thereof is envisaged as residing on such a workstation.

In another preferred embodiment the system is deployed on one or more servers remote from the LAN or WAN of the user where the workstation used by the user runs, instead of on the user's own computer, in a virtual client environment on one of those servers. The graphical user interface generated by the system is transmitted to the user over the internet and user input transmitted back to the virtual client to provide command and control of the virtual client session. This embodiment has the advantage of performance since the
network traffic over the internet is minimised with all data processing undertaken locally on the servers. A further advantage is that the computer required by the user needs only to be able to receive and display the interface transmitted by the server and in turn return the user's input.

Thus when viewed from a further aspect the invention provides an information management system comprising a client machine and a server machine in data communication with the client machine and adapted to generate a plurality of user interfaces on the client machine from a common interface structure, comprising means for retrieving data essential for the generation of each of said interfaces from a configuration data storage means on said server machine.

Preferably one of the client and server machines comprises user operable means for modifying the data stored in said configuration data storage means.

When viewed from a further aspect the invention provides an information management system comprising a client machine and a server machine in data communication with the client machine, said client machine being adapted to display a plurality of user interface instances which differ from each other under the control of reconfigurable data stored on said server such that any interface can be transformed into any other by altering said reconfigurable data.

The invention extends to a server and server software comprising means for storing reconfigurable data and adapted to supply said data to a client which is adapted to display a plurality of user interface instances which differ from each other under the control of said reconfigurable data supplied by said server such that any interface can be transformed into any other by altering said reconfigurable data.

The invention also extends to a client and client software adapted to receive data from a server and to display a plurality of user interface instances which differ from each other under the control of said data received from said server such that any interface can be transformed into any other by altering said data.
When viewed from another aspect the invention provides a method of developing an integrated information software system comprising providing a core system including means for generating a plurality of user interfaces, and storing data in interface configuration data storage means, said data controlling the appearance and functionality of said respective user interfaces.

When viewed from a yet further aspect the invention provides a business process for developing an integrated information software system comprising the steps of specifying a customer's requirements and entering said customer requirements in the form of configuration data to configure a core system to provide said integrated information software system, wherein said requirements are in at least one category selected from the group comprising: the appearance of a user interface; the functionality of a user interface; and the available routes with which user interfaces can be navigated by a user of said integrated information software system.

The embodiments of the invention described here permit the rapid development and deployment of computer-based IISS's each of which meets the differing requirements of a different purchaser, in an improved manner compared with current methods.

Underpinning the design is the analysis that while there are marked similarities in the underlying data structures of the existing ESS requirements of different organisations, the user implementations of each existing IISS requirement are different and in many cases unique.

The invention can provide the opportunity for the creation of a business process which first ascertains the IISS requirements of an organisation then delivers an IISS matching that requirement but does this an improved manner compared with the existing means of developing and supplying IISS's.

The Inventor has recognised and taken advantage of the similarities of such underlying structures. This allows for rapid development of the unique elements, enables rapid development and deployment of IISS's designed to meet the specific needs of a huge and diverse range of SMO's.
Using existing computer hardware, operating systems and applications, the Rapid Development Environment ("RDDE") provided in accordance with preferred embodiments of the invention enables:

(i) a Publisher/Developer or Configurer to rapidly deliver an IISS to a client's specification; and

(ii) a Publisher/Developer to rapidly create and deploy additional functionality.

The preferred embodiment includes suites of interfaces, tools and services, described below, enabling the process of specifying and configuring a system to function, for the:

(i) Publisher/Developer: (configuration and additional functionality);

(ii) Configurer: (configuration);

(iii) Administrator (some configuration); and

(iv) User (configuration of their own user options).

The speed and accuracy with which configuration and additional functionality can be delivered by a developer using preferred embodiments of the invention offers a substantial competitive advantage to software publishers and developers compared to their peers building an IISS by traditional means.

In its preferred embodiments the Invention permits a business utilising them, whether a publisher, developer or configurer, (in any case a "Developer") to deliver to their SMO clients the IISS that the client requires in a timely, efficient and profitable manner. Further, the specific modular design of the core system enables:

(i) the Developer to efficiently and profitably support their client with easy to install updates; and

(H) the Developer to apply the configuration module previously configured for an existing client for a new client with similar requirements and then adjust the configuration as required and deliver the new client's specified IISS quickly and profitably.

The Administrator of an IISS built using the preferred embodiments of the invention has at their disposal a series of interfaces, tools and services enabling them to perform their role quickly and efficiently.
Users of an IIS built using the preferred embodiments of the invention receive a visually informative, interactive, consistent presentation of their information enabling them to work more efficiently. This means that users naturally wish to undertake work within such a system. Working within it is seen as an advantage rather than a burden. In consequence, such an information system efficiently and contemporaneously collects data derived from the end-user's work, providing up-to-date information derived from that data to other end-users and the administrator.

BREIF DESCRIPTION OF THE DRAWINGS

A preferred embodiment of the present invention will now be described, by way of example only, with reference to the accompanying drawings in which:

Figure 1 is a schematic block diagram of the main parts of a system embodying the invention;

Figures 2(a) to (g) are block diagrams illustrating a non-exclusive range of possible file structures for the main system components;

Figure 3 is a schematic block diagram illustrating the use of stored and environmental data in controlling the operation of an interface button within an embodiment of the invention;

Figure 4 shows an implementation of an embodiment of the invention broken down into four phases and further broken down into eleven separate stages; Figure 5 provides an illustration of some of the tools that may be used by the embodiment in each of the 11 stages presented in Figure 4;

Figure 6 gives an example of the indirection structure behind the software used by the tools presented in Figure 5;

Figure 7 further shows various interconnections between the principle components, their division into Interfaces, Tools and Services as well as the principle direction of the data flow through these interconnections; and

Figure 8 shows the components and connections used when a user tries to login to the system, controlling both data access and the interface screens to be displayed.
Embodiments of the invention will be described by first setting out basic principles of its operation, and then describing a novel integrated information software system (IISS) development method that is enabled by the invention with an overview of the structure of the embodiment. Detailed descriptions of the various parts then follow.

DESCRIPTION OF PREFERRED EMBODIMENTS

The components of an embodiment of the invention will now be described with reference to Figures 1 and 2(a) to 2(g).

An embodiment of the invention comprises various Components which may be arranged in many different ways as described more particularly below:

(a) A Component is a logical construct, not necessary a physical manifestation.
(b) Hence the Components can be physically arranged in many different ways and sub-divided or combined whilst still falling within this specification.
(c) For example, the System & User Data Component(s) may conveniently comprise multiple digital files stored or hosted on multiple different computers potentially running under different computer operating systems. This is not necessary but it is possible and may be desirable in some cases.
(d) Fig. 1 illustrates the six logical Components of an embodiment of the invention: the Configuration Component (1) in which is stored the configuration data controlling the Interface Component (3) providing the Licensee's Users with their specified integrated information software system (ITSS); the Specialist Screens Component (2) stores particular User interface screens specific to each Licensee; the Deployment Licence (4) stores key data controlling and informing the IISS both so that it is only useable by the permitted Licensee and controlling access to the licensed features and modules; the System & User Data Component(s) (5) store data controlling use of the IISS and the User Data that the IISS has been configured to store and deliver to Users; the Admin User Accessible Logic and Interface Component (6) is accessible at Developer Level to higher level Users employed by the Licensee so that they can develop their own layouts, reports, logical structures and other elements without recourse to the Developer,
(e) Each Component may logically communicate with any other Component(s), shown by the arrows linking Components but it is preferred in any arrangement where
some Components are located on a server accessed by means of a network that such Components on the server cannot communicate directly with those on a workstation, that they are merely responsive to requests and instructions directed to them by Components on any workstation. This is because in such a case it is generally preferred that the Components located on the server should be able to function normally on their own without requiring the presence of or access by means of a network to a set of Components located on a workstation.

Fig. 2(a) illustrates a digital computer file structure where the components (1) - (6) have all been assembled as parts of a single file

Fig 2(b) illustrates a digital computer file structure where the components (1) - (6) have been assembled as parts of two files, one the Interface File (A) to be accessed by any User (containing Components (1) (2) & (3)), whether it is located locally on a workstation or remotely by means of network communication, the other, the Server File (B) (containing Components (4) (5) & (6)), which may be located remotely on a server or another workstation or which may be located on the User's own workstation.

Fig. 2(c) illustrates a digital computer file structure where the components (1) — (6) have been assembled as parts of three files. This is similar to that described in fig 8.1(b)(ii) except Components (1) and (2) have been separated out into a Configuration File (C) located locally to the Interface File (A) which contains the Interface Component. This has the advantage that a User can merely replace the old Interface File with a new version in order to use improvements contained in the new version, since the new version will use the existing configuration stored in the separate Configuration File (C).

Fig 2(d) illustrates a digital computer file structure where the components (1) — (6) have been assembled as parts of four files. This is similar to that described in Fig. 2(c) except Component (6) the Admin User Accessible Logic and Interface Component has been located in a separate Business Logic File (D). This has the advantage that the Server File (B) can be replaced on an upgrade whilst the Licensee retains Component (6) in the separate File (D) containing their own in-house development work which will continue to function in conjunction with the new File (B) containing Components (4) and (5).
Fig. 2(e) illustrates a digital computer file structure where the components (1) - (6) have been assembled as parts of five files. This is similar to that described in Fig. 2(d) except Component (2) the Specialist Screens Component has been located in a separate Specialist Screens File (E). This has the advantage that the Configuration File (C) can be replaced on an upgrade whilst the Licensee retains Component (2) in the separate File (E) containing their Specialist Screens and Layouts which will continue to function in conjunction with the new File (C) containing Component (1) and the existing File (A) containing Component (3).

Fig. 2(f) illustrates a digital computer file structure where the components (1) - (6) have been assembled as parts of six files. This is similar to that described in Fig. 2(e) except Component (4) the Deployment Licence Component has been located in a separate Licence File (E). This has the advantage that the Licence File (E) can be replaced in order to provide the Licence with revised Licence Terms without accessing or changing any other part of the IISS.

Fig. 2(g) illustrates a digital computer file structure where the components (1) - (6) have been assembled as parts of multiple files. This is similar to that described in Fig. 2(f) except that every Component is shown as comprising multiple Files. For example: the System & User Data Component (5) could access a variety of different data sources hosted at different locations and possibly running under different computer operating systems.

In order to enhance the speed or response of the IISS to user interaction, any dataset stored on the server may be downloaded in whole or in part and stored temporarily on the workstation in order to be used and possibly amended on the workstation without suffering the delay inherent in network communication with the server. Records amended in this way on the workstation may be uploaded back onto the server, so replacing the original copies of those records on the server, in order that information entered on the workstation shall thereafter be stored on the server. Record Locking and Validation procedures may be used to govern which record is stored after two users edit the same record simultaneously.
In order to operate the IISS the modules in the core system must retrieve and apply data from the appropriate configuration data storage. The particular example of the generation of a graphical element of a user interface screen is explained below in detail with reference to Fig. 3.

Fig. 3 shows a schematic representation of control of the display of an interface button, and thus whether it is active or not, providing to a User, to whom it is visible, access to a Function. A Function may be any action performed on behalf of or for the benefit of the User. This example illustrates a type of preferred method that may be used extensively in an IISS in accordance with the invention.

The use of "Buttons", digital graphic metaphors for physical "Buttons", is a common technique amongst computer interface designers. The Master Interface Model includes one or more of such "Buttons". In a preferred embodiment there may be many such buttons on each screen or layout.

The improvement given by the embodiments described herein over previous methods of controlling the display, and thus the use, of such Buttons (1) by the current user is achieved through the use of multiple layers of calculations which both control what each button looks like and whether or not each button is visible to the current user and thus whether the current user can use the Function accessed by that Button.

Whatever screen is displayed to the User, each object in the Interface forming part of the Master Interface Model, may be recalculated and redisplayed by this method whenever the User has some Interaction with the Interface.

By this means the same Interface Components may have a different Visual Appearance determined by a Calculation (11) informed by:

(a) The Interface Button (2) the subject of the Calculation (a Stored Resource)
(b) the current Data Module (3) in Use (an Environmental Variable)
(c) the current Active Mode of the IISS (4) (an Environmental Variable)
(d) which is the most recent Button (5) to have been Clicked or Selected by the User (an Environmental Variable)
(e) What Graphical Representations (6) of the Button and its various states have stored in the Configuration (a Stored Resource)
(f) What Textual and/or Graphical Icons (7) and what various states of each have been stored in the Configuration (a Stored Resource)

Further, whether or not this Visual Appearance is actually visible to the User (12), and thus whether or not the User can Access the Button in order to Click or Select it and thus call or run the Function (14a) (14b) contained in the Function Library (10) attached to that Button in the Configuration, is determined by

(a) Whether or not the Button has been Routed in the Layout Router and whether or not the Button has been allocated a Text/Icon (8) indicating its Function to the User - which determine whether or not the Button is notionally Active (Stored Resources)
(b) The Access Rights of the Current User (9) (compared with the Access Rights required to use the Function (10) called by the Button noted below) (Stored Resources)
(c) Other Environmental Tests (15)

The effect or benefits of this method are that

(a) A single Interface screen or layout may be used for many Users with different Access Rights, so that the higher level users will be able to see and use a wide variety of Buttons but various lower levels of users will only be able to see and use the more limited set of Buttons which they are permitted to use. This is preferred behaviour to offering the User many Buttons where when they try to click the button either it does nothing or they are told that they are not permitted to use this function.
(b) As previously described a copy of the Master Interface Model Screen may be used as the basis for any System Interface Screen in the IISS, the visual appearance of the screen is controlled by both Stored Resources forming part of the Configuration and dynamic Environmental Factors resulting from the User's interaction with the Interface, thus
   (i) Reducing development time when creating new interface screens and
   (ii) Ensuring absolute consistency of interface for the User.

Every Button may have a Specified Function (14b) assigned to it in the Configuration. The Specified Function refers to a Function stored in and called from the Function Library (10).
To call the Function the User must Click or Select the relevant Interface Button (1). The User cannot Click or Select the relevant Interface Button if it has been rendered invisible by reason of its Calculated Status (12) being 0.

The embodiment herein described comprises four main parts: a core system which includes basic 'shell' structures for the functions of the IISS; configuration data which is used by the core system structures to generate the IISS; user data which is the data which the IISS acts on as required by users; and a rapid development and deployment environment (RDDE) which manipulates the configuration data. The RDDE is only provided to those users who will be altering the IISS itself, e.g. configurers or developers, as opposed to end users who will only work with user data.

An exemplary method of developing an integrated information software system in accordance will now be described with reference to Figs. 4 to 8.

Figure 4 shows a process of using an embodiment of invention broken down into four phases and further broken down into eleven separate stages.

In the first phase 101 the requirements of the customer or vertical market for the IISS are analysed and a specification is agreed. In the second phase 102 the IISS is configured to meet the requirements agreed in phase 101. In the third phase 103 documentation is prepared and in the fourth phase 104 data are loaded. The process defined by these four phases can be further broken down into eleven separate stages, corresponding with the four phases.

The first more detailed stage 105, corresponding to the first overall phase 101, is to obtain information defining the customer's requirements for an IISS. (Numbers shown within boxes in Figures 5 and 6 in brackets correspond with components shown in Figure 7 where they are not specifically numbered in this description.) The next seven stages correspond with phase 102. They are: to configure a copy of the most recent version of the core system with the customer's details 106; to configure a licence file to suit the customer's requirements 107; to set up the users of the system in accordance with the customer's requirements 108; to set up fields to meet the customer's requirements 109; to
set up interface screens and printable screens and reports to meet the customer's requirements 110; to set up functions to meet the customer's requirements 111; and to create a new integrated information software system 112.

The next two stages 113, 114 correspond to the third phase 103; they are to prepare documentation 113 and to prepare training material 114. The final stage 115, corresponding with the fourth phase 104, is to import data provided by the customer and process them as necessary.

Figure 5 provides an illustration of some of the tools that may be used by the method of the invention in each of the eleven stages presented in Figure 4. For obtaining information defining the customer's requirements 105 the specification tool 16 is used to capture data about fields, reports, users and other requirements. To configure a copy of the most recent version of the core systems with the customer's details 106, the settings inspector 17 is used. To configure a licence file to suit the customer's requirements 107, the licence details are entered in an empty secure licence file 18.

To set up the users of the system in accordance with the customer's requirements 108, the access authority tool, confidentiality level tool and EDDT management tool may be used as at step 19.

To set up fields to meet the customer's requirements 109, the specification tool may be used to provide screen labels for appropriate fields together with the language inspector and the case tool as at step 20.

To set up interface screens and printable screens and reports to meet the customer's requirements 110, the following modules 21 may be used: the single master interface model; the layout component inspector; the standard fields inspector; the interface inspector; the language inspector; the layout router inspector; the category attribute tool; the user defined lookup tool; the calculation inspector; the window tool; the button parameter tool; and the column sort tool.

To set up functions to meet the customer's requirements 111, the following modules 22
may be used: the button parameter tool; the function library; the calculation inspector and the workflow tool.

For preparing documentation 113 the help tool, button help tool, and layout help tool (23) may be used.

Figure 6 gives an example of the indirection structure behind the software. It shows how the tools, inspectors and models of Figure 5 are linked to various tables and other data structures. The settings inspector for setting up customer’s details (17) uses the settings tables (43). For entering licence details in an empty secure licence file (18), the licence tables (44) are used. The access authority tool (24) uses the user’s tables (45) and the EDDT tables (46). The confidentiality level tool (25) uses the user’s table (45), while the EDDT management tool (26) makes use of the EDDT tables (46), the sales tax tables (47), the transaction codes tables (48) and the currency tables (49). In a similar manner the arrows show further links between other models, tools and inspectors and the various tables and data structures that they use. In detail, these are: the graphic elements and tiles tables (50, 51); the layout components tables (52); the language tables (53, 54); the standard fields key tables (55); the new dialogue cases tables (56); the multi-channel marker tables (57); the new record cases tables (58); the router tables (59); the window tables (60); the button parameter tables (61); the column sort tables (62); the function library tables (63); the user definable data calculations tables (64); the user definable look up data tables (65); workflow tables (66); workflow-stages tables (67); and the help tables (68).

Figure 7 shows a Schematic View of the Principle Components of the Software Element of the Invention and their General Types of Interconnection. The components are split into those that run on the Workstation (69) and those that run on the Server (70). The Workstation contains Interfaces for use by the Publisher/Developer or Configurer (71); Interfaces by use by the Administrator (72); and Interfaces for use by the User, to the extent allowed by the Account (73). The Workstation further contains various Tools for use by the Publisher/Developer or Configurer (74) and tools for use by the User (76). There are also services available to the Publisher/Developer or Configurer (77) and services for the User (79).
The Server (70) contains a Deployment Licence (18) which for example stores data controlling the licensee and their users' use of the IISS in accordance with the licence granted to the licensee by the Publisher / Developer / Configurer including which user data modules are available; the number of users who may be given access; the number of users who may log in simultaneously; the number of enterprise IDs available; whether or not multiple personalities are permitted; and whether or not multiple locations are permitted. The server also contains tables and tools for use by the Administrator (80) and tables for use by the Publisher/ Developer or Configurer (81). The server contains data modules for use by the User (82).

Figure 7 further shows various interconnections between the principle components as well as the principle direction of the data flow through these interconnections. For example, the Layout Router Inspector in the interfaces for the Publisher/Developer or Configurer (71) exerts a principle control over the services for the User (79). As a further example, the User Preferences Tool in Box 76 stores its data in the User's Tables on the Server in Box 45 which controls various aspects of the user interface in Box 73.

Figure 8 shows the components and connections used in logging a User into the System, Controlling Data Access, and controlling the Interface Screens to be displayed. During a Workstation log in attempt (83), the User Name and Password entered are checked, via the Network (86), against the User Data (45) stored on the Server (70). If the User is determined to be valid, the Licence validity is then checked against the Licence Data (18) stored on the Server (70). If the Licence is valid, processing proceeds to the next step, in which data are uploaded to the Workstation after log in (84). The Workstation is then logged in to the Core System as an Authorised User (85). Components of the logged in Workstation (85) use the network in the logged in state (87) to access components of the Server (70) such as Data Modules, User Defined Calculation Tables, and User History.

The logged in Workstation (85) contains controls for the Interface appearance, controls for determining which interface screen is to be displayed, and controls for accessing data. The User and Licence Data uploaded from the Server (45) in Step 84 includes inter alia the EDDT Membership Details; the User's Access Level, Function Level, Module Access Level, Data Authority and Confidentiality Level; the Current Enterprise Data; and Licence
Data. These data are used by the logged in Workstation (85) for determining what the User sees within the User Interface after User Interaction is completed. For example the Licence Data read from Server (70) during the Log in attempt (83) and passed through the uploading stage (84) to the Workstation logged in (85) are used to determine what permitted modules the User can access, which in turn is used to control the data displayed.

A more detailed explanation will now be given of the features of an exemplary RDDE and general features available to an IISS created thereby, including the features referred to briefly above.

General Structures and Use of Standard Terms
Data in the RDDE and the Core System, in common with any IISS, is stored in Tables, each of which contains zero or more Records (or Lines), each of which contains one or more Fields (or Columns).

Fields may be of different types and are used to store different sorts of data.

Different types of fields behave in different ways, for example Text Fields and Numeric Fields.

In order to display more than one Record in, for example, an AlphaNumeric or Date order, Records said to be Sorted by whichever Field or Fields are used as the Sort Key.

In order to find Records containing certain data in certain fields, or a range of such data, a set of Records are said to be Filtered from the total set of Records on some specific criteria.

Access Rights and Levels
The Developer Level user has access to all components in the "Publisher/Developer or Configurer" and the "Server" areas illustrated in Figure 7 together with everything available to both Administrators and Users. Figure 7 is described in greater detail later.

The Administrator Level user has access to all components in the "Administrator" and some in the "Server" areas illustrated in Figure 7 together with everything available to
Users.
The Users only have direct access to the components in the "User" area illustrated in Figure 7 and through those components to user data (82) on the server (70). In order to provide a greater granularity of control of users and flexibility in their management to administrators every user is allocated a series of numeric values, stored in their user record on the server, which control the interfaces, tools and data they are permitted to access. Those described below are exemplars of this method. Sets of access rights may be grouped and allocated to one or more users as an account type.

Access to Interface Elements:
(a) Each User is allocated an "Access Level" a numeric value controlling which specific layouts within the user interface of the IISS ("User Interface") they will be permitted to access.

(b) Each User is allocated a "Function Level" a numeric value controlling the visibility and thus availability of every interface button calling a function from the function library (264).

(c) Each User is allocated a series of "Module Access Rights" numeric values controlling their access to each user data module (82).

Access to Specific Data Records:
(a) Each User is allocated a "Data Authority" a numeric value which is accessed by the Access Authority Tool in determining to what data they should be permitted access.

(b) Each User is allocated a series of "Confidentiality Levels", a numeric value which determines the Confidentiality Level of the records that User is permitted to view. A Confidentiality Level can be attributed to any record in order to control access.

(c) Each User is given specific EDDT membership(s) as described above which control what data they are permitted to access.
Command and Control
Any user may interact with the RDDE and/or an IISS created using the RDDE by a variety of means including:

(i) telling the computer "pointer" to click an object in the interface, which itself invokes some response from the computer, by using a mouse or other pointing device, a touch sensitive screen or voice recognition; and
(ii) telling the computer to respond using Keyboard key strokes or voice recognition.

Control of the Interface and Data Display Structures, and Dependencies of the Core System
The Core System uses a series of Environmental Factors to control which User Interface is displayed, what are the state or appearance of graphical elements in that Interface and what data is displayed to the current User.

An example of this Dependency Structure is illustrated schematically in Figure 8.

Developer Interfaces, Tools and Services
Publisher/Developer/Configurer Interfaces (71)

Layout Inspector
A copy of every User Interface Layout or Screen used by all levels of User throughout the Core System is stored with relevant data within this Inspector (201).

This enables the history of each Layout to be recorded and accessed so that, for example, a Developer can check which Layouts are actually used in a particular Deployment when planning to upgrade specific aspects of the Core System.

Status Inspector Plus
The Status Inspector Plus (202) displays the current values of significant system variables to a Developer.

This aids debugging optimisation and development work by permitting the Developer to observe changes in system variables during a process.
Settings/Licence Inspector Plus
The Settings/Licence Inspector (203) displays the current values of system settings and licensee settings. System settings enable or disable Services and/or features within those Services. Licensee settings display, for example, the name of the Licensee, numbers of permitted users, enabled Data Modules and Features.

Access to these values enables:
(i) a Developer in the case of System Settings to change the behaviour of the IISS, for example controlling the circumstances in which a User History Record is created or whether or not a higher level User can Unlock a Locked record in each Data Module; and
(ii) a Developer in the case of Licensee Settings to change, for example: which Data Modules are accessible; how many Users can use the IISS; how many Enterprise Ids are permitted; whether Multiple Personalities are permitted; and whether Multiple Locations are permitted.

In the case of Licensee Settings in a normal Deployment only a Developer with a very high level of access to the Deployment Licence on the Server can change these settings but in other deployments Developer access through this Inspector may be permitted.

Layout Component Inspector
The screens or layouts used by all levels of users throughout the Core System are assembled from layout components stored in tables accessed through the Layout Component Inspector (204) ready for use when building new screens or layouts.

Components can merely be dragged and dropped or copied and pasted into their required locations on new or amended screens or layouts.

This Inspector stores data on each Component enabling the Developer to search for Components fulfilling specific criteria.

Standard Fields Inspector
Each Data Module contains a very large number of "Standard Fields" of each different
data type which can be used by the Developer for a specific client system build without needing to add new fields to the RDDE and Core System structure to match the client's requirements.

An important principle is that the structure of the RDDE and the Core System that results fully accommodates the requirements of every customer within either standard structures fields and functions, which are merely relabelled to suit each customer in their interface, or through use of indirection to provide for such needs within the standard fields, the Calculation Inspector and the User Defined Lookup Tool ("UlU Tool") further described below.

The Standard Fields Inspector (205), and the Table to which it provides access, records all the standard fields' module, field name and other details and enables the Developer to record what each standard field has been used for. Generally the client's own field data name or label is recorded.

The Inspector also displays the Text Label used in the User Interface (stored as a Language Variable in a Language Table) to label the field enabling the Developer to edit that Text Label as required.

The Inspector includes a variety of Filters and Sorts to enable the Developer to quickly and easily locate the Standard Field records to which he wishes to refer and/or edit.

Interface Inspector

The Interface Inspector (206) enables every element within the User Interface to be independently configured using graphical and/or textual elements. The data tables containing the Interface data can be saved and reloaded into any other version of the IISS or replaced with any other set.

This enables the Developer to give different client's users different interfaces in terms of colour, contrast and language.

Thus, for example, users with specific visual disabilities or requiring specific language,
whether in terms of desired ethnicity of language or language of work area, can be catered for.

The interface elements can be categorised as follows:

(i) "Graphical Tiles" used as Tabs, Buttons, Labels, Icons, Symbols, Dividers and Backgrounds. These may include a variety of different versions for each single interface element which enable its "state" to be shown to the user through visual changes in appearance;

(ii) "Text Strings" used to label Tabs and Buttons as an alternative, or in addition, to using Graphical Labels, Icons or Symbols;

Thus a trial system can be quickly set up by merely typing in textual strings to label parts of the User interface as required which can latter be replaced with high quality Graphical Labels Symbols or Icons in the deployed version.

The Interface Inspector enables the Developer to edit every interface element in order to independently control the appearance of each part of the user interface.

Event Inspector

The Event Inspector (208) and the Event Tables, to which it provides access for inspection and editing, contains a table of records recording specific events that occur during use of the IISS (Event Record(s)) and tables of records containing information about each type of event (Explanatory Tables).

The Records created become a Log of significant events during the use of the IISS.

Events may record, for example, system errors or administrative or user actions.

In case of a System Error, each Error Record records any relevant information concerning an error event including when where and how it occurred together with error code key(s) linking to records in the Explanatory Tables which enable interpretation of the error codes.

The Error Records enable the Developer and those providing System Support to identify
and analyse procedures within the IISS which require development and/or potential changes in training materials and documentation.

In the case of Administrative Actions, these Records may record occasions when, for example, the Administrator creates or disables a user, deletes a user's history or takes other significant steps in changing the data held by the IISS.

In the case of User Actions, these Records may record occasions when, for example, a User deletes a record or takes other significant steps in changing the data held by the IISS.

Layout Router Inspector

A user navigational step within the IISS is instigated by the user through User Command and Control and the Layout Router then determines the destination Interface Screen / Layout which is then displayed to the user.

The identity of the next screen to be displayed is controlled by the BP Key (defined below under Button Parameter fool) derived from the user's last User Command and Control interaction with the IISS. The BP Key gives access to a record in the Layout Router configuration tables storing the identity of the screen to be displayed. These tables can be edited by the Developer using the Layout Router Inspector to direct the user to any screen required at any point in the user's navigation.

The Layout Router Inspector (209) can be used to configure many different data tables. One configuration may provide access to Interface Screens / Layouts within the RDDE which are never accessed by another configuration. The Developer can substitute one table or set of tables with another set to provide the client's users with a completely different navigational and interface experience.

Each Layout Router data table stores the relevant data for a user's navigation of the IISS in a single record. Each table has multiple records. Each record stores Layout Router data required for a single user access level. Multiple records store the Layout Router data for multiple user access levels. Accordingly, the Layout Router can be used to control the user navigational experience of many different user access levels. Each user access level can
have, if required, a completely different IISS user interface.

The Layout Router results in all user navigation around the IISS being "soft-coded", thus editable by changing the value of a variable, without accessing the underlying coding. This gives the Developer great flexibility. If a client wants a particular Layout Screen to be displayed to a specific level User at a certain point in their interaction with the IISS the new screen is produced, based on the Single Master Interface Model described above, and the Router set to display that screen at the specified point in the user's future navigation of the HSS.

There is a substantial security benefit from this approach to user navigation. Since all User interaction with the IISS is controlled by the Layout Router there is no possibility of flawed logic within a complex system of controls as can otherwise occur within other types of Information System, resulting in a User accessing an inappropriate screen.

Publisher/Developer/Configurer Tools (74)

Window Tool

The Window Tool (210) and the Window Table, to which it provides access for inspection and editing, contains records in which are stored variables controlling the display of User Administrator and Developer Interface Screens (Window Variables).

Each Window Record contains a unique key identifying the Screen to which the record relates.

When the Layout Router has specified the Screen to be displayed the position, scale and size of the Screen are controlled by the Window Variables contained in the Window Table record for that screen linked through the unique key. On closure of the Screen the Window Variables are updated so that next time the Screen is used it will be displayed where the User last placed it, with the same size and scale.

Non-exclusive examples of other data which can be stored in the Window Table are variables identifying the type of Screen, controlling access, controlling whether or not the Screen is printable and if it is the relevant print variables such as paper source, size,
orientation and scale

Button Parameter Tool

The Button Parameter Table ("BPT") configured through the Button Parameter Tool (211) provides the Developer with great flexibility when building a new IISS within the Invention's RDDE enabling the behaviour of the IISS and its interface to be changed without requiring any access to the underlying code.

User Command and Control interaction with the System Interface gives access to a system of software commands and functions which enable navigation to be undertaken, records displayed, created deleted or manipulated in various ways described elsewhere in this specification. Each virtual button within the User System Interface stores a single numeric value. This value combined with an Environmental Variable derived from the current context of the IS, provides a unique key ("the BP Key") giving access to the appropriate record in the BPT. The values stored in this table provide control over the IISS Interface and thus the user experience. The Interface can be set to change its appearance and behaviour dependant on specific values obtained from the current BPT record accessed by the unique key.

The Button Parameter Tool enables the Developer to edit the variables stored in each record of the BPT in order to independently configure the behaviour of every Button within the interface.

Further, the BPT may also be used as a storage medium by programmed procedures running within the ESS ("Programs"). Such a procedure may output a change of value or values held in a specific record or records of the BPT in order to change the way in which the IISS responds to user interaction in the future.

The BPT also enables the programmatic Function called by specified User Buttons to be selected and set from the list of available functions.

The variables stored by the BPT can be categorised as follows, all of which include variables used by Programs and/or written to by Programs:
(i) variables controlling the activation or deactivation of individual Interface Elements (such as Buttons); or
(ii) variables required to control other aspects of the user interface and behaviour of the IISS.

Specification Tool
The Specification Tool (16) enables a client with or without the help of a Developer to record their principal IISS requirements. The data stored by the Tool may be used manually as a definitive information source by the Configurer, forming an instantly documented specification and/or to automatically configure an ESS within the Invention's RDDE.

The Tool assists the client in organising data taken from legacy systems so that it can be automatically imported into the new ESS.

The Tool enables the data fields ("fields") required by the client to be defined and linked to the existing fields within the standard data structure of the RDDE.

The Specification Tool can then be used to define the different screen layouts of fields that they require for records, listings, reports, outputs and other layouts, to define the fields on which the user must be able to search and what is to be displayed when records have been found and to define any other process or control that they require within their new ESS.

Help Tool
The Help Tool (214) and the Help Tables (68), to which it provides access for inspection and editing, enables the creation and updating of both User and Developer help records.

Help Table records may store textual Help together with other Help resources such as images and movies which explain or elucidate use and operation of the IISS.

Help may relate to any specific object on any Interface Screen / Layout and/or any Interface Screen / Layout.
Help records each contain a unique key identifying the screen or interface object to which they relate. Accordingly, the Help is contextually sensitive. The User or Developer is provided with the appropriate help for the screen or object in which they indicate their interest.

Help may non-exclusively comprise any explanatory material including explanatory text, images, animated "movies" with or without a sound track which may comprise a commentary, web pages viewable through a web browser.

Thus, for example, Help may explain what a specific interface buttons does or may explain how to undertake an entire process.

Any user may also obtain additional Help explaining concepts, terms or words used in Help text.

**Case Tool**

The Case Tool (215) (58) and the Case Tables, to which it provides access for inspection and editing, automatically creates and records the correct relationships between user data records within the IISS in an unlimited number of different circumstances.

Every data record within the RDDE contains a series of fields ("Data Key Fields") by which it can be related to one or more records in any other data table(s). Thus any record can be related to any other record(s). The data required to create these relationships can be automatically recorded on record creation and/or at any later time. This is controlled by the settings in the relevant record in the Case Table. An Environmental Variable automatically provides the Case Key to access the correct Case Table record.

The Case Table contains records each of which controls the automatic creation of relationships between the subject record and any other records in the IISS in specific circumstances.

The appropriate Case Table record controls which Data Key Fields in the current record
are populated with the unique keys of other records at the point at which the Case Table Control is invoked.

The Developer can use the Case Tool to create new Case Table records, and to edit existing Case Table Records, in order to control automatic relationship creation where appropriate.

Column Sort Tool
Every column of data displayed in the form of listings within the IISS can be configured to sort alternately A to Z and Z to A for different types of data for example text number date and time data by configuring the Column Sort Table on the Server

Each record in this table relates to a specific User Interface Layout. The names of the Column Headings are stored in this Table.

If a new Layout containing any type of listing display is created a new record is created in the Table, and the type of listing, name of the layout and the identities of the fields in the listing entered.

The new layout will then provide full Column Sort services.

The records created and configured through the Column Sort Tool (216) are stored on the Server in the Column Sort Tables (62).

Publisher/Developer/Configurer Services (77)
Single Master System Interface Service
The Invention uses a single "Master" computer layout screen which is copied and used as the foundation for every user System Screen throughout the IS. Every element in this screen automatically changes to reflect the type of data it is being used to display, the current Active Mode within which it is being used and the availability of user functions. In consequence the user receives a very consistent and responsive experience (263).

This enables the Publisher/Developer to create new system screens very easily, by merely
copying the Master screen, and adding Data layout components, ensuring complete accuracy and consistency.

The Interface can be built to change its appearance dependant on the values it derives from the Button Parameter Table, calculations using its own name and other variables such as the current Data Module and user inputs ("Environmental Variables").

Thus, for example, the RDDE offers a number of different "Active Modes" for the IISS each of which changes the behaviour and appearance of the current copy of the User Interface Screen. The Active Mode of each User Interface Screen is noted within its technical name. Thus when the Layout Router is directed to display a new layout the active mode to be used for that layout is determined by information incorporated within its name.

The Master Interface layout screen comprises many graphical and textual fields in multiple layers. The graphical or textual objects displayed by each field, at any given point in the use of the IISS are controlled by a calculation engine which itself is controlled by Environmental Variables.

Copies of the Master Interface can be used for virtually any purpose throughout the IISS subject only to minor configuration.

Administrator Interfaces and Tools (72)(80)

Status/Licence Inspector

The Status Inspector (221) displays the current values of significant system variables to an Administrator.

This aids debugging and development work by permitting the Administrator to observe changes in system variables during a process and provide appropriate information to those providing Support Services.

Settings Inspector

The Settings Inspector (222) displays the current values of system settings and licensee
settings. System settings enable or disable features and services. Licensee settings include, for example, the name of the Licensee and numbers of permitted users.

An Administrator can observe but not edit Licensee Settings which control, for example, the number of permitted users and the Data Modules which can accessed.

However access to System Settings does enable an Administrator for example to:

(i) change the behaviour of the IISS. For example the Administrator can set, in respect of each Data Module whether or not Users with a higher access level are permitted to Unlock Records which have been Locked; and
(ii) set the correct paper sizes and orientations for ease of printing

Language Inspector
The Language Inspector (207) and the Language Tables, to which the Language Inspector provides access for inspection and editing, contains records in which are stored textual variables to be displayed on User and Developer Interface Screens ("Language Variables").

Administrators have access to the Language Inspector in order to amend language as they wish.

Each Language Record contains many Language Variables any of which may be used on any Screen within any Interface.

All textual material, not comprising Data stored in Data Modules on the Server, which is displayed to users (non-exclusively including textual labels, screen messages and status messages) may be stored in a Language Table.

An IISS may be conveniently localised for any language or market by substitution of one or more Language Table or Tables with replacement(s) in the desired language.

Users/EDDT Inspector
Developers and Administrators accessing the Users' Table through the Users/EDDT
Inspector (225) can create, edit, enable, disable, and delete Users. They can also interrogate the User's history data on the server in order to check what any specific user has been doing.

Each User's record in the Users' Tables (45) can be edited to change variables such as their Name, Password, Access Level, Data Authority, Function Level, and EDDT Team Membership. The latter is explained further below.

No user can access any part of the RDDE or an ESS built using the RDDE, other than the login screen without entering the correct user name and password at login in circumstances where their user table record has currently an enabled status.

The Users/EDDT Inspector can be used to inspect and edit a User's Team Membership.

The Users/EDDT Inspector (225) provides access to a User's Team membership. Every Team forms part of a hierarchical structure within multiple layers of the structure of an Enterprise.

There can be as many levels to the structure as required.

For example, in a four level system, every Team belongs to a Department, every Department to a Division, and every Division to an Enterprise.

Each User can belong to as many Teams as required.

The Team or Teams of which a User has membership is/are used by the Access Authority Tool to determine the User's access rights to any given record.

The Users/EDDT Inspector is used to inspect, add or remove the Team memberships of any User.

The Users/EDDT Inspector provides an Administrator with access to the EDDT Management Tool (235) and Tables (46) on the Server.
Calculation Inspector
Since every different client/deployment will require different calculations to be performed on numeric data, even within the same "Vertical Market" (organisations working within the same sector and jurisdiction) the Calculation Inspector (224) gives the Administrator direct access to the formulation of such calculations in each Data Module. These are Used Defined Calculations, which may, if required, utilise data stored in the User Defined Lookup Tables (65).

The mathematics controlling each calculation field is available to be edited as a text string resulting in either a valid result or an error message and brief explanation of how the text string edited by the Administrator fails to meet the required syntax.

The User Defined Calculation Tables (64) are located on the Server and accessed from the Administrator Interfaces through the Calculation Inspector (224).

There are two main benefits: first the Client maintains direct control over the way in which their IISS calculates values numeric, textual or graphical and is thus able to adjust them over time as circumstances dictate and second the Client's Server installation can be moved to a more recent version by normal data transfer maintaining the integrity of their IISS without any further manual updating being required.

The Client can, of course, choose to build their own data structures within the Business Logic File which remain their own full responsibility and which will not be updated by the Developer when other parts of the IISS are updated.

Access Control/Authority Tool
The Access Control / Authority Tool (231) provides a very flexible and powerful means of controlling who can or cannot exercise each type of Data Access to any given Data Module record.

Possible types of access to a Data Module (together "Access Right(s)") include the right to:
(i) create a record;
(ii) view an existing record;
(iii) edit an existing record; and
(iv) delete an existing record.

Every record is owned and controlled by the Team or Teams of which the Record creator was actively a member at the moment of creation.

Every user may belong to one or more Teams.

A user belonging to more than one Team can, at any time, elect to be treated as currently inactive in one or more of the Teams to which they belong. Failing such an election they are treated as being concurrently Active in all their Teams.

Whether or not a user can exercise each of the preceding types of Data Access right is controlled by an Authority calculation.

The system administrator can adjust the following values which enable the Access Control system to be adjusted to meet user requirements:

(i) Authority Enhancements - every user permitted to use the IIPP has a base authority of 1. Each user can be given one of a number of enhanced levels of authority, each of which multiple their Authority by an enhancement value. The Administrator sets the numeric enhancement value for each enhanced level of Authority);
(ii) Authority Reductions - when a User's Authority is assessed in order to determine their access rights it is reduced, divided, by a numeric value for each of the Enterprise's Hierarchical levels that their current Team Membership(s) do not share, are separated from, with the Team ownership(s) of the subject record. The Administrator sets the numeric reduction to be applied for each level of separation; and
(iii) the Authority Required to exercise Access Rights in each Data Module - the Administrator sets the respective levels of authority required to create, view, edit or delete a record in each Data Module. Thus more sensitive or commercially important data may
be given a higher required Authority before a user is permitted to exercise an Access Right.

Thus every User has a Current Data Authority stored in their User Record comprising (1 x their various Authority Enhancements)

Actual Data Authority: Before any Access Right may be exercised, the user's Actual Data Authority is calculated as follows for a four level hierarchy (and similarly for however many levels have been used in practice):

\[
\text{Actual Data Authority} = \frac{\text{Current Data Authority}}{\text{Authority Reduction: separation from Data Record Enterprise} + \text{Authority Reduction: separation from Data Record Division} + \text{Authority Reduction: separation from Data Record Department} + \text{Authority Reduction: separation from Data Record Team}}.
\]

This calculation refers to the current EDDT Model of the Organisational structure, rather than that which existed at the date of creation of each record enabling organisational structural changes to be easily catered for.

Whether or not the User can exercise the Access Right in question depends on whether their Actual Data Authority in this case is greater than or equal to the Data Authority required to exercise this Access Right as set by the Administrator as described above.

Confidentiality Level Tool

This tool (232) provides the System Administrator with considerable flexibility in controlling which users can see which information. It enables the System Administrator to give each User a different Confidentiality Level in relation to any Data Module.

The different Confidentiality Levels required are setup by the Administrator in the UIu Table and then the relevant levels are allocated to each User.

The Administrator can create edit and delete Confidentiality Level Records in the UIu table at will.
Users with the ability to create and/or edit Data Module records can set the User Confidentiality Level required to have any access to each individual record.

Users have no access to any Data Module record where their own Confidentiality Level is lower than the Level required for access specified in the record.

EDDT Management Tool
Whilst the EDDT Inspector (225) enables the Administrator to check and change a User's Team Membership so the EDDT Management Tool (235) enables the hierarchical structure of the Enterprise to be modelled ready for use in the EDDT Inspector

This Tool is located on the Server and accessible from the EDDT Inspector.

Each record belongs to one of the defined hierarchical levels. For example in the case of a four level system a record is created for each Enterprise, each Division, each Department and each Team. Each record, depending on its level, is specified to be a "child" of the appropriate "parent" record. Thus Team Z may be the child of Department Y, itself the child of Division X, itself the child of Enterprise W.

When the Client Business is (inevitably) reorganised, the relevant changes to the EDDT model of the business structure can quickly and easily be made to enable all future Data Access Rights to function as the newly reorganised IISS Users would expect.

Enterprise Identities Inspector
The RDDE incorporates the ability to service multiple Enterprise Identities simultaneously administered through the Enterprise Identities Inspector (223) and the Multiple Enterprises IDs Tool (237) on the server.

Subject to this being permitted by the Deployment Licence an Administrator can create more than one Enterprise Identity in order to concurrently run multiple enterprises.

The Administrator can configure many different Enterprise Identities stored in the
Enterprise Tables (238) so that a single IISS can provide all required services both to users working for different separate Enterprises or for users working for more than one Enterprise at the same time.

5 Configuration of an Enterprise Identity includes all the data that distinguishes that Enterprise from any other, for example, data comprising its:

(i) Corporate Identity - logos, names, design elements, registered number(s);
(ii) "Letterhead" - address, telephone etc.;
(iii) Tax Identity - Registered No., Taxation Periods; and
(iv) Financial Identity - Accounting Codes, Conventions, Accounting Periods.

The hierarchical structure of the Enterprise is then modelled using the EDDT Management Tool accessible on the Server.

15 Categories and Attributes Tool
This tool (239) enables the System Administrator to create, delete and edit the Category Attribute records in the tables (240) from which fixed Category and Attribute value lists are created.

20 The Administrator can thus create appropriate Categories for use in each Data Module then create the required list of Attributes to be displayed in relation to each Category in each Module. Each set of Categories and Attributes may be set to be either Fixed or Dynamic.

25 A dynamic set permits users to add values to either Categories and/or Attributes which are then reflected in future lists of available Categories and Attributes. A Fixed set means that only those values entered by the Administrator using this Tool will be available.

30 User Defined Lookup Tool
This Tool (241) enables the System Administrator to create, delete and edit User Defined Lookup Table (65) ("UIu Table") records which are both used to create User Defined Value Lists and to store Used Defined data to be used by User Defined Calculations.
Thus values stored in the UIu Table can be amended as necessary, for example, where fulfilling a Rate Card function, to reflect changes in the Enterprise's pricing structure. New records can be added, amended or deleted to ensure that the Administrator has complete control over the data used by the Value Lists and Calculations configured to use this data.

The effect of this UIu structure is to ensure that any unique values used by the client enterprise within calculations are stored within the data structure of the RDDE completely separate from the system structure.

User Interfaces and Tools (73)(76)
Consistent User Interface
The Interface (263) presents to the User a variety of graphical and textual items providing information regarding both the Data in which they are interested or with which they are working and the Command and Control options that are available to them as a User.

What the User sees in respect of both the Interface and the Data is controlled by a series of dependencies based on Environmental Variables as illustrated by way of example in Fig 8 where the Interface presented is controlled by the Layout Router informed by the user’s Access Level, the state of elements in the Interface is controlled by data from the Button Parameter Table, the user’s Function Level and Module Access Rights and the Calculated Active Mode derived from the Layout Name whilst the Data actually displayed to the User is controlled by a combination of Environmental Variables including the User’s Data Authority, Confidentiality Level, Team and Enterprise membership and Deployment Licence Data.

The consistent interface enables users to quickly learn how to use any IISS built using the invention.

The use of the Single Master Interface Model by the Developer ensures that consistency is quickly and easily achieved and maintained.
Users find that consistency in the interface quickly provides them with a feeling of familiarity which leads to confidence and pleasure in using the TLSS.

The user undertakes different activities in different Active Modes but is not aware of this fact. The Active Mode is one of the Environmental Variables controlling the appearance and behaviour of the user interface. The basic Active Modes are Browse, Edit, Find and Print, but other Active Modes can be created by the Developer as required.

Mirroring of Server Tables
In order to provide improved performance for the user some data tables, or sections of such tables, may regularly be copied from the server to the workstation Mirror Tables (290) for local use on the workstation. Such tables may comprise system configuration data and/or user data. Means may be provided for such data to be written back to the server and for conflicts resulting from the same record being edited by different users to be resolved.
In a preferred embodiment key system settings are set and stored on the server in the settings tables (43) and are thus editable by an administrator from any workstation. On login the user's workstation uploads these settings from the server for local use during the user's session.

Means may be provided to automatically upload a new set of settings from the server in certain circumstances.

Multiple Find Options Interface
Users of each different access level can be provided with a variety of different means of "Finding" or searching for records (261). Each means of finding records in each module uses a different screen design which can thus be tailored exactly to a client's requirement. This means that screen designs can be simple and easy to use rather than becoming congested with too many potentially confusing options.

The biggest issue for most users is being easily to find, retain, add to and reduce from the various sets of records with which they are concerned. Thus a variety of different user find screens can be offered to each level of user, each matching their specified need.
Each time the user invokes "Find" the last Find Screen that they displayed in that Data Module is redisplayed aiding continuity of work flow.

Records can be marked for instant recall using the Multi Channel Marker System ("MCM") described below.

When the User's find operation has concluded the set currently being displayed is transferred into Browse Mode for further attention.

An alternative find operation ("QuickFind") may be offered in Browse mode enabling the user to search on a small number of common fields and instantly display the record(s) meeting those criteria without entering Find Mode which may thus be reserved for undertaking more complex queries.

Further, the automatic record linking service controlled by the current configuration of the Case Tool means that often a user does not need to actually "Find" records. Merely by clicking on a linked record or Tab giving access to another Data Module will immediately display all the records in the target Data Module linked to the current record. Thus the user can often "find" records without having to undertake any "Find" selection as such - the operation becomes automatic. Since any currently displayed records can be instantly marked with the MCM and then instantly recalled, this provides great ease of use and flexibility for the differing requirements of different users.

Reports can be created in a variety of ways. The IISS is preconfigured to enable a user to view and print reports comprising records they have found by whatever means, whether pre-ordained or through their manual selection. The user can also dynamically specify their own report contents, within predefined formats and save those hybrid reports for future reuse.

Users have access to any programmatic Functions specified by the Developer by clicking the appropriate user interface button. They have no ability to change which function is called by which button.
PrePrint / Merge Tool
When a printable object requires printing the PrePrint / Merge Tool (265) is displayed. This provides a variety of features including selecting the normal print options (what is to be printed, number of copies, paper format etc) previewing that which is to be printed, setting the sort order and format of the printed output and outputting to other formats such as PDF other software program formats and email.

The Merge options within the Tool enable the user to set up merged output to dynamically include data from each record. For example to personalise the contents of a letter to include reference to information held within the IISS in relation to the intended recipient or job or project to which the communication relates.

User Preferences Tool
The User Preferences Tool (266) gives each user access to settings controlling how the IISS responds to their use. For example these could include:

(i) EDDT membership - when a User belonging to Teams in more than one Enterprise logs in they are asked to elect which Enterprise for which they are currently working. At any time a User with more than one team membership can elect under which Team Membership or Memberships they are currently working. This controls both the team Ownership of records that they create and their current Data Authority;
(ii) working for another user - any user can elect to be currently working for another user at any time. Such a user retains their own Access Authority, thus their Authority cannot be enhanced by this means, but records that they create are deemed to have been created by the person for whom they have elected to work. In consequence, such records become data over which the person for whom they elected to have been working, and that person's colleagues, will have the Authority that they would have had had that person actually created the records. However, the one difference from the case where such records were actually created by the deemed creator is that the actual creator retains access rights to such records but such rights can be varied or revoked by the deemed creator. The benefit of this feature is that staff providing, for example, secretarial support to professionals do not thereby acquire ownership of records which logically should be
owned and controlled by the professional responsible for the work; and

(iii) work location - the User can specify their work location at any time so that that information can be automatically included with other data they are recording.

(iv) which of a series of different interface options the user wishes to use

(v) the perspective the user wishes to use for automatic linking to different records

(vi) whether user history is created

(vii) whether helpful dialogues are displayed

(viii) whether the mirroring of key data tables on the workstation from the server is updated at login.

Button Help Tool

A single click of an interface button switches on Button Help (267). Until this Help is turned off clicking on any button in the User Interface displays information about what that button does. This allows the user to find out what each button does before actually using its functionality. This information can be printed. The user can turn off Button Help and revert to normal operation with a single click.

Layout Help Tool

A single click of an interface button displays the Layout Help Tool (268) giving information about the layout currently displayed, describing its functions and how to use them.

Keywords within this Help information are themselves explained with additional Help displayed by clicking on the keyword. This aids user understanding and results in the Help data being non-repetitious and thus easier and quicker for the Developer to update as further versions of the IISS are released. This information can be printed.

Properiser Tool

It is common to provide users with some means of automatically capitalising the case of the first character of names. The Properiser Tool (269) provides users with an easy means of dealing with unusual and anomalous use of letter case and symbols in words whilst retaining the convenience of automatic conversion of names to the correct case.
This issue occurs primarily in relation to person, organisation, and trade names where non-standard use of upper and lower case characters make automatic correction of case difficult.

The tool enables users to specify:

(i) words that should be left exactly as they were entered; and
(ii) words that should always be capitalised.

The first character of words which have not been specified under either special category will then automatically be capitalised when the Properiser function is applied to a specific field.

User Services (79)

These comprise the range of key services delivered to every user, subject to their user account status, in an identical manner within every data module.

Basic Navigation

Within any set of records the user can always move through the set sequentially using the Previous or Next Services or move to one end or other of the set using First or Last Service.

Create a New Record

Users having the necessary Access Rights may create new records. New record creation may be part of an automatic process or may be "manual".

In any event the new record may be linked at creation to other records to which the current record on which the User is located is related through the current configuration of the Case Table and the Case which the IISS is configured to call at this point.

Where the new record creation is manual a New Record Dialogue is displayed enabling the User to select from existing records, from any relevant Data Modules, if this appropriate, to which to link the new record.
Where new information is inserted in the Dialogue which does not relate to any existing record the IISS will automatically create another new record as necessary.

For example, if a new Communication is created addressed to a Contact without a record in the Contact Data Module a Contact Data Module Record for that individual will be created automatically as the new Communication is created.

Edit a Record

Subject to a User's rights enabling them to View any given record they may or may not be entitled to Edit that record.

If they are entitled to Edit a record they will be given access to the Editing section of the User Interface, otherwise they will have no access.

Duplicate a Record

As an aid to rapid working a "Duplicate" of an existing record can be made ready to be changed as necessary rather than creating a whole new record from scratch.

For example, a user wanting to enter an Expense may find it convenient to find a previous similar example, duplicate that, and change it as necessary.

A Duplicate record is not a true duplicate. It is a new record superficially similar to the original record but with its own unique identity.

Delete a Record

With the appropriate access rights a user may delete a record. The procedure is designed to ensure that this cannot be undertaken unintentionally. On trying to delete the record the user is prompted to hold down another key whilst clicking the delete button again.

Generally, only the Administrator has the ability to delete more than one record at a time.

Select a Record
The select button reduces the current set of records to just the current record.

Omit Record(s)
The Omit button omits the current record and <ah> Omit displays all the currently omitted records. This provides a very useful combination of functionality when dealing with sets of records.

Highlight Current Record on a listing
Record Listings automatically highlight the current record with a contrasting colour background for ease of reference.

Automatic Record Linking
The Developer has configured the Case Tool so that Data Module records are automatically linked to any other relevant records in the same or any other Data Modules. This enables the User to always be able to quickly and easily display sets of records which are relevant to their current task or requirement.

A refinement to automatic linking may be to provide two or more different perspectives for the basis on which linking is established and/or the basis on which it is displayed. For example whilst viewing a history record with both a related job and a related contact the user may wish to link automatically to communications relating to the job or to the contact - each requirement comprises a different perspective.

Sorting all Columns A-Z or Z-A
All Listings can be sorted A-Z and Z-A by clicking the column heading. This service is controlled by the data stored in the column sort table record for any given screen.

Categories and Attributes
Every record can be marked with a virtually limitless variety of Attributes. These can be selected from pre-defined fixed lists, capable of being edited by the system administrator, to control the input, where users are not permitted to enter a value not already in the list, or dynamic so that they automatically offer all the Attribute values already entered in that Data Module and permit the user to add new values not previously entered.
Every Attribute is grouped under a single Category under a single Data Module. Thus the user, wishing to select an Attribute for a record must first select the appropriate Category and then select the desired Attribute from the various Attributes listed or grouped under that Category within that Data Module.

Users can select and record as many Attributes as they wish in respect of every record. These Attributes can then be searched for or found using a multi layered Boolean logic filtering system where, for example, records with Attribute "A" and Attribute "B" or Attribute "C" or Attribute "D" (and for many more iterations of the and/or logic) can be instantly displayed and then marked using the Multi Channel Marker described above. The Attributes filtering can be combined with numeric and/or data range filter criteria.

Back/Forward (through User History)

The users path through the records of the IISS is recorded in a user history table on the Server.

This enables the User to go backwards or forwards over the records they have previously reviewed, whether in this user session or a previous one.

On logging in the user may be automatically placed on the record on which they were located when they last logged off and thus can work in a continuance from their previous session.

This results in there being no requirement for a main menu or start screen.

This also enables the Administrator to review who has been doing what were. Thus actions taken by a user can be stored in the User History, providing a comprehensive audit trail where required.

Coding of Records

Coding in its basic form enables a set of records to be marked with an alpha-numeric string as an identifier to enable that set of records to be recalled at any time.
An interface may be provided to enable records to be found by reference to their coded string.

Any set of records so coded is known as a List

Manage Lists
A further interface, for List Management, may be provided whereby records within a given coded set may be recalled and then records added or removed from that set by means of a procedure that adds or removes the current code from one or more specified records.

Users with appropriate rights have access to interfaces enabling them quickly and easily to add records to a coded list or remove records.

Manage Parenting
Any record can be marked as the "parent" of any other record where a Data Module is configured to provide this service. This enables the construction of "Tree Structures" where records can be linked easily and quickly to a series of other records through an number of intermediate stages.

For example Organisations control or ownership of other Organisations can be modelled using this Service, as can Contact Networks.

Users with appropriate rights have access to interfaces enabling them quickly and easily to add or remove Parenting relationships.

Manual Sorting of Records
Whilst all listings provide the column heading Sort Service described above all Data Modules can also be sorted by any combination of selected criteria required by the user.

Lock Records
Any records can be locked to prevent any future amendment. Such locking can be absolute
or subject to an overriding ability controlled by the System Administrator to unlock the record only able to be utilised by high level users.

Multi-Channel Marker System

Every record, in every Data Module, can be "marked" by a user in any one or more of several different user "Marker Channels". Each user can name their own marker channels as they wish. A single click will immediately display all records marked by this user in the current marker channel. Channels can be changed at will. The user can instantly display records marked in any combination of the seven channels. This powerful system makes it very easy for users to do what they want most, to be able to redisplay any given set of records they have been dealing with quickly and easily.

The user can also display records marked in more than one channel by merely selecting the MCM channels on which to search.

For example in an IISS with eight MCM channels the user may have access to seven channels and one can be kept hidden for system or Administrator use.

Server Deployment Licence, Tools and Data (70)(18)(80)(81)(82)

In a typical embodiment of the Invention the "Server" components described below would be located on a physically separate computer providing the Server service. However, this is not necessarily the case. The Server components could equally form part of a single deployment of the Invention on as single computer, used for example, by a single user as a workstation, or by a server offering multiple user "virtual" sessions to remote or network users.

The Deployment Licence

The Deployment Licence (18) is a preferred feature of the Invention. The Licence controls what the user is entitled to do in terms of both the Data Modules available for use by the user, date by which the Deployment Licence expires, what specific services the user may utilise and how many users may be authorised to use the IISS or may be permitted to login at one time.
The Developer is enabled, using these Licensing variables, to configure a Deployment of the πSS in whatever is the most commercially appropriate manner for a given client.

For example, a client wishing to lease the IISS can be given an Expiry Date a specific period after the next Lease Payment is due, and be sent a new Expiry Date code after payment has been received.

Events Tables
These tables (246) store the records regarding Events automatically created throughout the Core System which can be inspected by an Administrator from any workstation through the Event Inspector.

Case Tables
These tables (58) store the Case Records created by and edited through the Case Tool which control the automatic creation of relational links between Records in different Data Modules.

Column Sort Tables
These tables (62) store the Records created and configured through the Column Sort Tool.

Access Authority Tool
This Tool (231) enables an Administrator to edit a User's Authority to access Records in each Data Module.

Confidentiality Level Tool
This Tool (232) enables an Administrator to edit a User's Confidentiality Level which controls which Records in each Data Module they can access based on the comparison between their Confidentiality Level and the Confidentiality Level of the individual Record.

Users Tables
These tables (45) store a Record for each User which is created and edited by an Administrator through the Users/EDDT Inspector.
Users History Tables

These tables (234) store all the Records created by each User as they navigate through the IIS if that feature has been activated by an Administrator. This data being stored on the Server means that a User can log on to any Workstation and use their existing User History to move back or forwards over the records over which they have navigated during a previous session.

EDDT Management Tool

This Tool (235) enables an Administrator to create and edit multiple hierarchical layers in the structure of an Enterprise, to which Users may be given membership using the Users/EDDT Inspector.

EDDT Tables

These tables (46) store the EDDT Records created and edited by the EDDT Management Tool.

Multiple Enterprise IDs Tool

The RDDE incorporates the ability to service multiple Enterprise Identities simultaneously. The Administrator can configure many different Enterprise Identities using the Multiple Enterprise Ids Tool (237) so that a single IIS can provide all required services to users working for different separate Enterprises or for users working for more than one Enterprise at the same time.

Configuration of an Enterprise Identity includes all the data that distinguishes that Enterprise from any other, for example, data comprising its:

(i) Corporate Identity - logos, names, design elements, registered number(s);
(ii) "Letterhead" - address, telephone etc.;
(iii) Tax Identity - Registered No., Taxation Periods; and
(iv) Financial Identity - Accounting Codes, Conventions, Accounting Periods.

The hierarchical structure of the Enterprise is then modelled using the EDDT.
Enterprises Table
These tables (238) store the Enterprise Records created and edited by the Multiple
Enterprise IDs Tool.

Category and Attributes Tool
This tool (239) enables the System Administrator to create, delete and edit the Category
Attribute records from which fixed Category and Attribute value lists are created. The
Administrator can thus create appropriate Categories for use in each Data Module then
create the required list of Attributes to be displayed in relation to each Category in each
Module. Each set of Categories and Attributes may be set to be either Fixed or Dynamic.

A dynamic set permits users to add values to either Categories and/or Attributes which are
then reflected in future lists of available Categories and Attributes. A Fixed set means that
only those values entered by the Administrator using this Tool will be available.

Category and Attributes Tables
These tables (240) store the records created and edited with the Category and Attributes
Tool as well as the Records created by Users' use of the Category and Attributes Service

User Defined Lookup Tool ("UIu Tool")
This Tool (241) enables the System Administrator to create, delete and edit User Defined
Lookup Table (65) ("UIu Table") records which are both used to create User Defined
Value Lists and to store Used Defined data to be used by User Defined Calculations (64).
Thus values stored in the UIu Table can be amended as necessary, for example, to reflect
changes in the Enterprise's pricing structure. New records can be added, amended or
deleted to ensure that the Administrator has complete control over the data used by the
Value Lists and Calculations configured to use this data.

The effect of this UIu structure is to ensure that all unique values and business logic
calculations used by the client enterprise are stored within the data structure of the RDDE
and thus as data, may be moved from one version of the IISS to another. Hence this UIu
structure divorces the licencee's unique business logic from the formal database structure
of the of the ESS.
User Defined Lookup Tables
These tables (65) store the records created and edited with the User Defined Lookup Tool used to store values used in Value Lists and User Defined Calculations

User Defined Calculations Tables
These tables (64) store the records edited through the Calculation Inspector (224) which give an Administrator the ability to define their own calculations with results to be displayed in the User Interface

Workflow Tool
An Administrator can use this Tool (244) to create and edit Workflows, which are series of Tasks to be undertaken in relation to, for example, a Project; which a User can then use, via a Function, as the model on which to base automatic creation of a whole series of Tasks to be performed or completed in relation to, for example, a specific Project.

Within the user interface a specific workflow can be selected and applied to any other record so that a series of Tasks then exist to be undertaken in relation to that record. If the workflow is designed to be sequential this can enable management of processes automated exception analysis based on what tasks have or have not been completed.

Each Task can be configured by the Administrator to be created only if specific values relating to, for example the related Project, hold a specific value or are within a certain range. Thus the creation of individual records in the workflow can be intelligent so that only relevant Task records are created.

Workflow Tables
These tables (66)(67) store all the Workflow records and record sets, created by and edited with the Workflow Tool. Record Sets enable individual records within a workflow to be grouped into Stages.

Other Administrative Tasks, Processes, Tools and Tables
Other tasks, processes, tools and tables (260) for the Administrator may be created on the
server when required.

User Data
User Data Modules (82) comprise the core building blocks of the user data holding aspect of the Core System. Each Module contains data that is essentially different to every other. Some records are Transactional meaning that one or more such records may be created primarily in relation to a Non Transactional Record. For example, these could include:

- Contacts (Non-Transactional Records) - details of individuals each having one or more Personalities;
- Personalities (Transactional Records) - accommodates different Personalities for Individuals, for example multiple Home and Work Personalities without limit;
- Organisations (Non-Transactional Records) - details of organisations each having one or more locations;
- Locations (Transactional Records) - accommodates different Locations for Organisations, for example multiple Geographic Locations of different offices or business units of the Organisation without limit;
- Communications (Transactional Records) - comprising Outgoing and Incoming Communications regardless of Format, thus for example emails, letters, faxes, memos, SMS's and file notes;
- History (Transactional Records) - details of every event relating to any other record, for example details of telephone conversations, meetings, consultations and actions which may relate to for example Contacts, Clients or Projects;
- Future (Transactional Records) - details of events or actions planned for some latter date in relation to, for example, Contacts, Clients or Projects;
- Clients (Non-Transactional Records) - many but not all Organisations have Clients, which may be individuals or corporate bodies, thus may be Contacts or Organisations;
- Projects (Non-Transactional Records) - a Project is a generic term covering something to be done for a Client, or for an Individual or Organisation where the user does not have Clients and includes something to be done for the user's own Enterprise - thus it is the place where most "things to be done" are separately identified in order that Transactional Records may then be created recording what is to be done (Future) or what has been done (Communications, History, Expenditure, Income and Ledger);
Lines (Transactional Records) - each Project can comprise an Order and the Line records comprise the line items in that order;

Records (Transactional Records) - a generic place to store any data or information not stored elsewhere, for example details of Physical Items such as Paper Files, Documents and other Storage Media or the address or location of any information, webpage or data such as a computer Uniform Resource Locator ("URL");

Products/Assets (Non-Transactional Records) - items to be sold, supplied or stored;

Movements (Transactional Records) — every movement of every Product/Asset is recorded to provide accurate data and a complete audit trail;

Bills (Transactional Records) - each record comprises a claim for payment, usually but not necessarily addressed to a Client, which may be in respect of a Project, and which is intended to create Income;

Expenditure (Transactional Records) - details of all expenditure allocated to relevant Projects and Accounting Codes;

Income (Transactional Records) - details of all Income allocated to relevant Projects and Accounting Codes; and

Ledger (Transactional Records) - double entry Ledger records created in relation to Bills, Expenditure and transfers of funds enabling the preparation of a Trial Balance, Balance Sheet, Profit and Loss Account, Cash Flow or any other relevant financial Analysis.

Possible Embodiments of the Invention

A Configurer can, using the Invention, specify, configure and deliver a high quality IISS exactly matching a client's requirement in an efficient and timely manner.

The Administrator of an IISS built using the Invention has at their disposal a series of Interfaces Tools and Services enabling them to perform their role quickly and efficiently.

Users of an IISS built using the Invention receive a visually informative, interactive, consistent presentation of their information enabling them to work more efficiently. This means that Users naturally wish to undertake work within such a system. Working within it is seen as an advantage rather than a burden. In consequence, such an Information
System efficiently and contemporaneously collects data derived from the User's work, providing up-to-date information derived from that data to other Users and the Administrator.

5 File Structure
A typical IISS built using the Invention's RDDE will include programming, other logical elements, configuration tables, data tables and user screens ("System Components") ready for deployment on the selected hardware under the selected operating system and utilising the support of the selected database application environment.

10 It may have a variety of different file structures dependant on the requirements of the relevant market and the services provided by the utilised database application environment. Possible embodiments include:

15 (i) a Single File IISS where System Components are included within a single file;
(ii) a Dual File IISS where the System Components are deployed on the user's computer Workstation in one file with the exception of the data tables which deployed on a network server in a second file (to enable, for example, reduced network traffic by undertaking most processing work and logical activity on the workstation leaving merely the data to be transferred over the network and enabling the workstation interface file to be upgraded by merely replacing it and importing the configuration tables records);
(iii) a Triple File IISS similar to the Dual File IISS but where the configuration tables are abstracted into a second file residing on the user's workstation (to enable, for example, the upgrading of the other file on the workstation, the User Interface and Logic file (excluding configuration) by merely replacing it with a new version);
(iv) a Quadruple File IISS similar to the Triple File IISS where a second file is provided on the server to which the users have a high level of access and into which they can build their own business logic enabling all network users to access their own Business Logic services whilst still accessing all the services provided by the rest of the IISS (to enable, for example, the upgrading of the data file by replacement and importing of data table records whilst leaving tile user's Business Logic file untouched);
(v) a Pentuple File IISS similar to the Quadruple File IISS where a third file is provided on the user's workstation to store specialist user screens designed and deployed
for this particular installation so that the rest of the system can be upgraded whilst leaving those screens untouched;

(vi) a Sextuple File IISS similar to the Pentuple HSS where a third file is provided on the server containing all the Deployment Licence data. This file acts as a Deployment Key without which the entire IISS is rendered inoperative; and

(vii) the many tables and other System Components comprising a typical IISS developed using the Invention subdivided and grouped in many different ways whilst still comprising an IISS within this specification.

Mode of Deployment
The method described above and illustrated in Figures 4-8 provides a means of deploying a high quality IISS using the Invention for a wide range of possible clients in many vertical markets which is quicker and results in a better quality outcome than is practicable with current methods.

Examples of Deployment
Some examples deployments of IISS created using the Invention may comprise:

(a) A Law Firm specialising in Intellectual Property Work who can operate with a support staff cost less than 25% of their competitors as a result of the business efficiencies delivered by such an IISS; such a system would have been tailored to meet their precise requirements, quickly and at a competitive cost;

(b) A Brand Management Firm providing brand management services to a client by way of a telephone helpline requiring a quick and simple means of recording helpline calls and reporting on those calls to clients;

(c) A Film Studio providing foreign language dubbing services to the makers of films and television programs wishing to manage their processes through a computerised system - moving from paper driven manually managed processes to an IISS precisely configured to their needs generating significant business efficiency savings improvements to reporting to both clients and their finance department;
A Software Development company requiring a range of sophisticated services and low administrative overheads but an efficient service to clients.

In each of these example deployments no available existing software provided what the business required but neither was a custom software development program acceptable because of both cost and uncertainty of the outcome. In each case the methodology made possible by the Invention could enable the business to deploy the IISS it required, quickly and at an acceptable cost and benefit from regular system upgrades in the future without having to commission custom development work. In each case the IISS deployed may be identical versions produced using the RDDE, merely configured differently to deliver a completely different IISS to each client.

It should be appreciated by those skilled in the art that the foregoing is merely an exemplary implementation of the invention and that the principles thereof could be implemented in markedly differing embodiments.
CLAIMS

1. An information management system comprising:
   an information database (5) operable to store data records (82);
   an interface component (3) operable to generate user interfaces enabling a user to
   input data and access and manipulate data records (82) in said information database
   (5);
   a context data store operable to store data identifying a user and the operations
   performed by a user; and
   a configuration store (1,2) operable to store configuration data identifying specific
   operations to be performed in response to user input when identified data is stored in
   said context data store;
   wherein said interface component (3) is responsive to user input to generate user
   interfaces enabling a user to input data and access and manipulate data records (82) in
   said information database (5) on the basis of said user input, context data stored in said
   context data store and configuration data stored in said configuration store.

2. An information management system in accordance with claim 1 wherein said
   interface component (3) is responsive to user input to generate user interfaces
   wherein the content of said user interfaces are determined on the basis of said user
   input, the context data stored in said context data store and configuration data
   stored in said configuration store (1,2).

3. An information management system in accordance with claim 2 wherein said
   interface component (3) is operable to restrict the operations which a user can
   perform by inputting data into a generated user interface on the basis of said
   context data stored in said context data store and configuration data stored in said
   configuration store (1,2).

4. An information management system in accordance with claim 2 or 3 wherein said
   configuration store is operable to associate items of content data with data
   identifying one or more users, wherein said interface component (3) is responsive
   to user input to select content data for incorporation in a user interface on the basis
   of said user input and context data stored in said context data store.
of the association of items of content data with data identifying a user stored in
said context data

5. An information management system in accordance with claim 4 wherein said
content data comprises any of graphics data, text data or sound data.

6. An information management system in accordance with any preceding claim
wherein said configuration store is operable to associate screen layouts with data
identifying one or more users, wherein said interface component (3) is responsive
to user input to select a layout of a user interface on the basis of the association of
screen layouts with data identifying a user stored in said context data

7. An information management system in accordance with any of claims 1-6 wherein
said interface component (3) is operable to select which of a plurality of different
operations to perform in response to user input when specified context data is
stored in said data store on the basis of configuration data stored in said
configuration store (1,2) associated with a user identified by said context data.

8. An information management system in accordance with claim 7 wherein said
plurality of different operations comprise different sort operations to perform on
data records (82) stored in said information database (5).

9. An information management system in accordance with claim 7 wherein said
plurality of different operations comprise different selection operations to perform
on data records (82) stored in said information database (5).

10. An information management system in accordance with any preceding claim
wherein said context data comprises:
data identifying a user; and
64

data (234) identifying one or more operations previously performed by said
information interface component (3) in response to user input associated
with said user.

11. An information management system in accordance with any preceding claim
wherein said context data store is operable to store data identifying groups of users
and said configuration store (1,2) is operable to store identifying specific
operations to be performed in response to user input when identified data
associated with an identified group of users is stored in said context store.

12. An information management system in accordance with any preceding claim
further comprising an administration module (6) operable to modify configuration
data stored in said configuration store (1,2) thereby modifying the operations
performed in response to user input when identified context data is stored in said
context store.

13. A carrier carrying computer interpretable instructions which when performed by a
programmable computer cause the programmable computer to become configured
as an information management system in accordance with any of claims 1-12.

14. A method of selecting operations to be performed by an information management
system comprising:
  storing context data identifying a user and the previous operations
  performed by a user;
  storing configuration data identifying specific operations to be performed
  in response to user input when identified data is stored in said context data
  store;
  receiving user input; and
determining the operation to be performed by said information management
system in response to the received user input on the basis of said user input, said stored
context data and said stored configuration data.
15. A method in accordance with claim 14 wherein determining operation to be performed by said information management system in response to the received user input on the basis of said user input, said stored context data and said stored configuration data comprises one or more operations select from:

- including one or more selected items of content with in a generated user interface;
- restricting the operations which a user can perform by inputting data into a generated user interface; and
- selecting the manner in which data records are filter or sorted.

16. A method in accordance with claim 14 or 15 further comprising:

- modifying said stored configuration data thereby causing said information system to perform a different operation in response to identical user input and context data.
Fig 1

1. Configuration Component
2. Specialist Screens Component
3. Interface Component: Contains the various interfaces which permit the normal user to interact with the User Data
4. Deployment Licence
5. System & User Data Component(s)
6. Admin User Accessible Logic and Interface Component

Fig 2(a)

1. Configuration Component
2. Specialist Screens Component
3. Interface Component: Contains the various interfaces which permit the normal user to interact with the User Data
4. Deployment Licence
5. System & User Data Component(s)
6. Admin User Accessible Logic and Interface Component
Fig 2(b)

Interface File

1 Configuration Component
2 Specialist Screens Component

3 Interface Component: Contains the various interfaces which permit the normal user to interact with the User Data

Server File

4 Deployment Licence

5 System & User Data Component(s)

6 Admin User Accessible Logic and Interface Component

Fig 2(c)

Configuration File

1 Configuration Component
2 Specialist Screens Component

Server File

4 Deployment Licence

5 System & User Data Component(s)

6 Admin User Accessible Logic and Interface Component

Interface File

3 Interface Component: Contains the various interfaces which permit the normal user to interact with the User Data

SUBSTITUTE SHEET (RULE 26)
1. IF the Current Button is Visible to the User (visibility is a metaphor for Availability) Clicking it enables the User to...

2. CALCULATED STATUS
   Control of whether any Button is Displayed to the Current User

3. CALCULATED APPEARANCE
   Control of the visual representation of the Button to be Displayed to the Current User - Subject to the Overriding CALCULATED STATUS

4. Specified Function is called by the Interface Button

5. STORED RESOURCES - Part of the Configuration
   - Has the Button been Routed and a Text/Icon Assigned?
   - Which Interface Button?
   - Graphical Representation(s) of the Interface Button
   - Textual or Graphical Symbol to be displayed on the Interface Button to suggest its use to the User

6. OTHER ENVIRONMENTAL TESTS
   - Data Module in Use?
   - Active Mode of the IISS
   - Which Interface Button has been Clicked or Selected?

7. CURRENT ENVIRONMENTAL FACTORS - dynamic values stored temporarily during User Interaction with the Interface
   - The Access Rights of the Current User compared with the rights necessary to Use the Function to be called by the Button
   - Other Environmental Tests
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>101</td>
<td>Analyse Requirement of Customer or Vertical Market for IISS and agree Specification</td>
</tr>
<tr>
<td>102</td>
<td>Configure IISS to meet the Requirement</td>
</tr>
<tr>
<td>103</td>
<td>Document</td>
</tr>
<tr>
<td>104</td>
<td>Load Data</td>
</tr>
<tr>
<td>105</td>
<td>Obtain Information Defining the Customer's Requirements for an IISS</td>
</tr>
<tr>
<td>106</td>
<td>Configure a copy of the most recent version of the Core System with this Customer's details</td>
</tr>
<tr>
<td>107</td>
<td>Configure a Licence File to suit this Customer's Requirements</td>
</tr>
<tr>
<td>108</td>
<td>Setup the Users of the system to accord with the Customer's requirements</td>
</tr>
<tr>
<td>109</td>
<td>Setup Fields to meet Customer's requirements</td>
</tr>
<tr>
<td>110</td>
<td>Setup Interface Screens and Printable Screens/Reports to meet Customer's requirements</td>
</tr>
<tr>
<td>111</td>
<td>Setup Functions Calculations and Workflows to meet Customer's requirements</td>
</tr>
<tr>
<td>112</td>
<td>New Integrated Information Software System</td>
</tr>
<tr>
<td>113</td>
<td>Prepare Documentation</td>
</tr>
<tr>
<td>114</td>
<td>Prepare Training Material</td>
</tr>
<tr>
<td>115</td>
<td>Import Data provided by Customer and process as necessary</td>
</tr>
</tbody>
</table>

SUBSTITUTE SHEET (RULE 26)