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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 March 2002 (14.03.2002)

PCT

(10) International Publication Number
WO 02/19790 A2

(51) International Patent Classification: Not classified

(21) International Application Number: PCT/US01/27721

(22) International Filing Date:
7 September 2001 (07.09.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
60/230,799 7 September 2000 (07.09.2000) US

(71) Applicant and
(72) Inventor: **BRIERE, Daniel, D.** [US/US]; 803 Warrenville Road, Mansfield Center, CT 06250 (US).

(74) Agents: **PETERSON, Thomas, L.** et al.; Banner & Witcoff, Ltd., Eleventh Floor, 1001 G Street, N.W., Washington, DC 20001-4597 (US).

(81) Designated States *(national)*: AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

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

WO 02/19790 A2

(54) **Title:** MARKETING COLLATERAL REPOSITORY AND SUPPORTING DATA MANAGEMENT AND COMMUNICATION ENVIRONMENT

(57) **Abstract:** A method and apparatus for collection and dissemination of information in an organized and systematic manner using a one or more computer networks. The information is collected from its originators, sorted and organized in an efficient and searchable manner. The information can then be used by buyers to research and choose the products and services that best fit their needs and wants.

MARKETING COLLATERAL REPOSITORY AND SUPPORTING DATA MANAGEMENT AND COMMUNICATION ENVIRONMENT

CROSS-REFERENCE TO RELATED APPLICATION

This application claims priority to U.S. Provisional Application Serial Number No. 60/230,799 file September 7, 2001 and entitled "Marketing Collateral Repository And Supporting Data Management And Communication Environment."

BACKGROUND OF THE INVENTION

The present invention generally relates to the field of information technology and more particularly, is directed to a method and apparatus for collection and dissemination of information in an organized and systematic manner using a one or more computer networks.

The World Wide Web's phenomenal success has brought a wealth of options to buyers' computer screens, allowing them to research and choose the products and services that best fit their needs and wants. In fact, so many options exist that searching for a product can easily become daunting. Information is difficult to find, conflicting, incomplete, or simply missing. The novice buyer becomes overwhelmed trying to discern who has what and what is important.

Increasingly, buyers are turning to trusted advisors (referred to herein as, "Influencers") to guide their purchase decisions. These Influencers come in many forms:

- People. Reporters, editors, financial analysts, industry analysts, associations/forums, regulatory bodies, etc.
- Web-based Media. Online vertical sites and buyer's guides covering a specific industry or a broad spectrum of technology-based products, directories, links pages, FAQs, comparative Web sites, and search engines.
  - Other Media. Print publications like magazines and newspapers, radio, and TV shows, such as MSNBC, CNET TV, etc.

Influencers fill a critical role in helping buyers to better understand product requirements and features and move more quickly to a purchase decision. They are the gatekeepers for Sellers to reach Buyers – whether buying optical telecommunications switching equipment or cleaning fluids for contract office cleaners. Influencers are the indirect channels that influence the mainstream purchasers. Sellers must, therefore, establish strong relationships with the many Influencers who have heavy influence over their Buyers.

There are many know Influencers in the purchasing process, including:

Web Content Publishers

- Vertical Web Sites/Portals
- Comparison Sites
- Search Engines
- Discussion Groups
- FAQs
- Links Pages

Print Publications

- Buyer's Guides
- Directories
- Analysts
- Industry Analysts
- Financial Analysts

Industry Associations/Forums/Regulatory Bodies

- Industry Standards/Trends
- Membership Directories
- Members Listed by Product/Services Categorizations
- Industry Statistics

Other Media

- Radio/Web Broadcast
- TV/Web Broadcast

SUMMARY OF THE INVENTION

At its core, the present invention is a massive database of product/service data, company and personnel information, contact information, personal preference information, event information, messages and communications, news, images and just about every other conceivable type of digital marketing collateral. The system of the present invention will alternatively be referred to as "mBlast".

The Gateway is a rich supporting layer of web-based applications for inputting, accessing, comparing, sharing, challenging, hosting and researching this data. The invention also offers a variety of supporting applications for the exchange of data such as APIs, pre-coded modules and hosted environments.

The methodology for the technical design of is to keep the data layer independent of the application layer as much as possible, meaning that the supporting applications are not a dependency for the data design and vice-versa. The data layer is architected for maximum flexibility in usage by a wide variety of applications, both now and in future releases and additions to the Gateway. The data layer is designed to allow for an infinite variety of data types, as more and more varied product/service categories are included and as the product matures in scope.

A key to the power of present invention is that every unit of data accumulated will allow a wide variety of information to be attached to it. A supporting data layer will contain meta and supporting data about each product/service/company data field, and will allow for expansion of the supporting data as the scope of the application grows.

The architecture of the invention uses a familiar n-tier web-application model. The back-end network consists of data warehouses, messaging servers, transaction servers, application servers, data exchange layers, and other supporting components connected to the internet. The middle-tier application layer consists of compiled application objects that communicate with the back end layer and the user interface. The user interface consists of, for the most part, web pages hosted on a public web server and dynamically generated from script-based templates. This architecture allows for maximum flexibility and scalability. Any piece, or entire layer, can be replaced or modified at any time without compromising existing functionality.

We have chosen a Microsoft-based platform for the initial release of the Gateway. This platform is robust, and offers a variety of tools that can rapidly increase the time to market for web-based applications. However the architecture is not dependent on the Microsoft platform, which allows the opportunity to migrate some or all of the application to another platform at any time.

Features

Gateway Environments

The Gateway is the web-based user interface portion of mBLAST . The Gateway is accessible via the World Wide Web, though registration is required for access by end-users. The Gateway consists of three main areas; data input, data output and data management. The Gateway provides a comprehensive collection of applications for performing these functions. The data management capabilities are provided in part a customizable end-user environment for leveraging the features of the invention .

Data Access– The Access Modules of the program allow for output clients to receive the data from the gateway. In the On Demand mode, users enter the system via a USERID/PASSWORD entry. They can view many different listings of the data, depending on their particular interest. Since the data is hierarchical (industry, product group, company, product line) users can click down to as discrete a listing as they require. Information is viewable in a listing format. There is also a matrix view format that allows users to either Select All or Select Specific Sellers to create a custom matrix; users have the option of selecting all possible fields for that category or selected fields as well. There is a transpose option that allows views with either features or Sellers across the top horizontal headers, with the Sellers or features in the vertical headers. Users can save data viewed as a file and send it as an email attachment. Data output modules include Buyer's Guide Matrix, Side-by-Side Comparison Engine, Company Profiles, Link Pages, Data Mining Engine, Glossary, etc.

Data Input – In this module, corporations can enter their data into predefined fields. Users enter the system via a USERID and PASSWORD-protected entry. Emphasis of the interface is on click boxes and radio buttons, with as little open text entry as possible to maximize the apples-to-apples comparison of the data. Each company/division has its own profile in the database core against which all data is attached. Corporations can select what industries and product categories in which they participate, and for each product category they are presented with additional screens for data entry, with most products having about 25 fields on average representing that product category.

My mBLAST – homepage for registered mBLAST users. Provides easy access to mBLAST features and personalized services, including messaging, "delta" data change notifications, automated data collection engine, etc.

mBLAST Gateway Supporting Applications

In addition to the central functions of inputting and extracting data, mBLAST Gateway offers supporting applications that leverage the data and invites the Seller and Advisor communities to interact with mBLAST.

Automated Supplemental Data Collection Engine

Several other tools will be offered to the input and output side clients that help create a set of capabilities that builds repeat visits and loyalty to the gateway itself. One of these is the Supplemental Data Collection Engine, which is an automated robot/spider that is constantly searching the Web for information pertinent to different tools on the site. For instance, PR personnel are likely to be interested in how their latest press release was covered by various publications, newsgroups, Web sites, etc. Specific press releases could be tied to a data collection spider that would visit regular sources of content for their coverage of the topic, driven by key words from the release. This engine could be used to drive competitive tracking capability for companies, or competitive article tracking tools for reporters who want to see how others covered their same story. This tool will be accessible throughout the gateway in different ways.

Messaging Engine

The Messaging Engine is a powerful, email-based tool present throughout the gateway's screens. On different screens it provides different support. For instance, the Data Access On Demand Module/Matrix Mode, enables the user to send a copy of the matrix shown to all public relations managers for each of the companies listed – so a reporter getting ready to run a story based on those facts can shoot the matrix off to the pertinent PR personnel for a final review before publication. The Data Access On Demand Module/Listing Mode, enables the user to send an email to all the PR personnel for the companies listed/checked, replacing many individual calls or emails to accomplish the same thing. Indeed, as the PR personnel register their names and areas of responsibility on the input side of the gateway, they enable a communications path to be created on the output side that makes the overall communications process far more efficient and direct. Future plans include enhancements to include functionality associated with instant messaging and unified messaging.

Data Validation Module

A sophisticated data validation module enables clients to double check data, to make use of the email engine to send data around the client's firm, and to perform side-by-side comparisons of products against the competition. In essence, this is a valuable competitive analysis and product management engine for marketing departments and product managers. Companies can enter proposed products online in a draft mode to see how they would stack up with current competitive products. Information is presented in an easy-to-understand matrix format. Companies can also enter data in advance and embargo the data so that it is only available to the public at a specific point in the future – such as would be required with a new press released product.

A key part of the data validation module – and where it draws its name – is the comment communications chain capability, which is visually demonstrated by the "Challenge" button. The Challenge button is an icon in the upper, right-hand corner of the screen that allows the present corporate marketing user to contest information presented by a competitor. So, if Company A believes that a cell of information from Company B in the presented matrix is not quite true, they can click on the Challenge icon and then click on the cell in question, and an interactive comment field is created. Company A would enter the rationale for questioning the data (the questioner, as initially planned, would be anonymous to Company B but not to mBLAST) and hit submit. An email is sent to 1) the designated PR contact for that product at Company B, and 2) mBLAST 's staff, for monitoring. The Challenge icon remains in the corner of the cell in question until a response from Company B is received by the gateway. Upon receipt of a response (notification of which is sent to 1) the questioner at Company A and 2) mBLAST 's staff for monitoring), the icon reverts to a Comments Available icon which permanently remains associated with that cell.

This feature is critical to the success of the mBLAST business model as it provides a mechanism for those who know most about those products – the Sellers themselves – to keep each other in check. Since this data is sent out to many different output clients, it becomes a critical part of corporate positioning to maintain watch over competitor data as well as one's own. It also relieves mBLAST of the burden of having to play middleman. Note that mBLAST is involved in the early stages of this process in a monitoring capacity; this is to watch for abuse and misuse of the feature. mBLAST may have to limit the number of times that this feature may be invoked to prevent some companies from questioning every single cell of every single competitor. (There is more discussion of this feature under Data Collection and Management heading later in this document.)

Importantly, these comment streams also provide Advisors with great angles for potential stories. In this sense, corporate positioning can be applied on top of mere data in this data stream.

mBLAST Support Applications and Value-Added Services

mBLAST also offers a variety of support applications and value-added services to clients and end-users, including:

Data Entry-Automated Push/Pull Module - If a firm already has product and corporate information databases, mBLAST 's gateway has open APIs that allow direct interface to the site for continual updating. mBLAST will accept any data that conforms to its data structure – using a record layout format publicly available on the gateway. mBLAST will accept data files in both tab delimited formats or XML coded formats. Direct, real-time feeds are also possible for select applications where it makes sense, such as with service availability (i.e., where service is presently available; this data tends to be highly proprietary, but makes sense to funnel through a gateway for ease of use). In this sense, the gateway might be "pulling" data from the client sites as opposed to them pushing data to the gateway.

Data Access-Automated Push/Pull Module – The direct push/pull module is more tuned towards self-feeding APIs. An output client might regularly download a new file to incorporate on its site or might choose to receive alerts from the gateway with "delta" records of changed data. Data is sent out in a specific format, and output clients are expected to create/adjust their programming to take advantage of these feeds. For the largest shopping portals such as AltaVista and Yahoo!, mBLAST will conform to their APIs for automated distribution of product information to these sites on behalf of its selling organizations.

Hosting – mBLAST will offer ASP-like hosting services, which will allow Web sites to include Buyer's Guides and other mBLAST modules in their Web sites without incurring the expense of site programming fees. mBLAST will work with customers to adopt the look and feel of their sites on its servers, providing a seamless experience for customers. mBLAST intends to resell the hosting services of its partners in providing these services.

Application Objects

The middle tier of the mBLAST architecture is a layer of application objects that comprise the business logic and perform the bulk of the Gateway functions. These objects are written in Visual Basic because of the quick implementation time allowed by the language. VB, in combination with a multi-system MTS environment allows excellent application scalability. For processor intensive objects or objects that may cause bottlenecks within the application, those will be developed in C++ or Java. Ideally the entire middle tier application object layer will be developed in C++ or Java a to allow for cross-platform scalability and to take advantage of the robust both-threaded architecture. The modular design of mBLAST will allow us to port the application objects to C++ or Java one piece at a time as is necessary.

The application objects will expose properties, methods and functions to allow control by the front-end scripting templates and by back end components such as databases, messaging servers, etc.

Data Access Objects - User Interface

Matrix Generator – accept user input to generate custom matrices of mBLAST data – ability to save Matrix designs in user account

Comparison Engine - The Comparison Engine module enables you to compare any range of companies and features against one another -- creating a more customized experience for viewers

Company Profiles – can be created/modified by form generator and linked to presentation templates

Link Pages - can be created/modified by form generator and linked to presentation templates

Data Mining Engine – sophisticated search capabilities across mBLAST data sets

Glossary – can be user defined or linked to from other sites

Data Packaging Module – Generate downloadable/email attachment file from screen data

Data Validation Module – allows a variety of supporting data to be attached to each data unit such as challenges and comments.

Data Spider Module – configurable search agent that spiders internet sources for pertinent data

Data Query Module – Coordinates database queries (searches, data retrieval, add/delete/update records)

Data Input Objects - User Interface

Form Generator – creates web-based forms based on pre-specified criteria (using the Administrator Form Creator)

Data Query Module – Coordinates database queries (searches, data retrieval, add/delete/update records)

Messaging Object

Features:

□ web-based email environment

□ instant messaging

□ unified messaging

□ file attachment capabilities

□ Fax/pager capabilities

□ smtp outgoing

□ pop3 incoming

Automated Data Input Modules

Automated Input Push Module – API connections to mBLAST data core for automated push delivery of data by input clients

Automated Input Pull Module – API connections allowing mBLAST automated retrieval of data from input client databases

Data Spider Module – configurable search agent that spiders internet sources for pertinent data

Automates Data Access Modules

Automated Access Push Module – API connections to output client databases allowing automated push delivery of mBLAST content

Automated Access Pull Module – API connections to mBLAST data core for automated retrieval of data by output clients

Hosting

The hosting service will leverage the User Interface Data Access Objects utilized by the mBLAST Gateway. An additional object layer will coordinate the administrative aspects of the hosting solution.

Hosting Administrative Objects:

- Hosting configuration module – could be coded into templates
- Branding module – coordinates the branding of the hosted solution to conform to the client site graphics/navigation etc.

Pre-Coded Implementation Modules

The pre-coded solutions will be script-based web-page templates that make remote calls to the mBLAST network for data and functionality. These modules can be created by packaging the mBLAST user interface templates into a generic format.

Pre-coded modules include:

- Buyer's Guide
- Events Listing
- Link Pages
- Glossary Listing
- Any HTML Presentation of data within the mBLAST system

Administrative - New Product/Service Definition Module

Need better understanding of the process to create a new product/service vertical before this module can be fully defined

- New Product/Service Creator
- Product/Service Hierarchy creator
- Data Field Creator
- Input Form Creator

Administrative – Content Management Module

This module allows for a cookie-cutter type approach to the user interface development. This allows all interface components to be mixed and matched to create generic interface templates, which then dynamically create the web interface based on user-based criteria (i.e. input form templates generate correct category input forms based on the category selected by the user).

There are a wide variety of off-the-shelf Content Management Solutions that could provide the needed functionality.

Web Templates

The user interface of the mBLAST Gateway is created with script-based web templates. This modular, template-based design allows for dynamic generation of web output based on the data set and the supporting application objects. This design offers several benefits;

Minimizes development time - Allows for reuse of templates for any product/service category without creating new code.

Keeps business logic out of web layer – logic encapsulated in the application object layer and Data Layer for better scalability and security.

Modular approach allows interface components to be changed/add/removed independently of other components without rewriting existing code.

Script-based front-end interfacing with application objects can leverage lower cost, and more plentiful developers for interface development than if the entire application was developed at the web layer (i.e. using scripting for all business and function logic).

mBLAST Gateway Template Areas

Following are the different sections that comprise the mBLAST Gateway environment;

mBLAST introduction – marketing environment for the mBLAST offering. This section will include:

Descriptive content

Web-based demonstrations of mBLAST functionality

Messaging

Registration – environment to register for mBLAST as either an output or input client

Log In – authorization component to allow entry to various mBLAST environments. Sets level of authentication for authorized users

Input Forms – data collection environment for Product/Service data

My mBLAST – customizable environment for utilizing the various data access and input components, messaging

Contact Companies – Messaging environment for communicating with specific company PR contacts

- Create/Edit Matrix – Data Access components interface – create or edit Buyer's Guide Matrixes, Side-By-Side Comparisons, Company Information, Links, Directories, etc.
- Edit Profile/Preferences
- Data Validation – sub-section of the Data Access Component interface – allows users to attach supporting data to existing product/service fields, embargo input data, etc.
- Messaging – standalone section (i.e. web-based email client) and also a sub-section of Data Access Component interface - allows user to email data, contact companies, etc.
- Hosted Environment – configurable environment utilizing some or all of the Data Access components

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features of the present invention are set out with particularity in the appended claims, but the invention will be understood more fully and clearly from the following detailed description of the invention as set forth in the accompanying drawings in which:

Figure 1 is a block diagram of the system of the invention;

Figure 2 – 8 are pictorial representation of various aspects of the present invention; and

Figure 9 – 11 are flow charts illustrates the operation of the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENT

In accordance with a preferred embodiment of the present invention, the system has a four-tier architecture.

User Interface

The first tier is the User Interface layer. This layer allows the user to input information into the system and displays information for the user to view. The UI layer is a combination of a web browser client and a web server. The web browser client can be any recent browser that supports HTML and Javascript (Internet Explorer or Netscape). The web server is Microsoft Internet Information Server, using Active Server Pages as a script language. The web server composes the HTML that is sent to the browser client, and interacts with the Business Logic tier.

Although the web browser is the primary UI, the system also supports a variety of other methods for getting information into the system and for retrieving information from the system, as illustrated in the diagram above. Information can be delivered via email, instant messaging, and in files in most file formats, such as XML. Information can be delivered instantly when information changes, via alerts, or batched and delivered on a scheduled basis.

Business Logic

The second tier is the Business Logic layer. This layer is responsible for encapsulating access to the database tiers, and for applying business rules, data validation, and business processes to the data when it is written to the database or retrieved from the database. The business logic layer interacts with the UI layer and the data/metadata layers.

The Business Logic layer is a set of COM applications written in Visual Basic.

Metadata

The third tier is the Metadata layer. This layer is a "data dictionary" that provides a definition of the raw data within the system, and stores rules for editing the data, retrieving the data into streams of information, and displaying the data in formatted presentations for the web and reports. The Metadata layer is also responsible for applying security to the data (restricting who can access and edit each unit of data). The Metadata layer interacts with the Business Logic and Data layers

The Metadata layer is stored in database tables in a Microsoft SQL Server database.

Data

The fourth tier is the Data layer. This is where the raw data is stored. Because the system defines the raw data using metadata, the raw data can be stored in a set of common tables. So different kinds of records, each with a variable number of fields and a variable number of values for each field, are all stored in the same tables. New records, forms, and datasets can be defined without making changes to the structure of the SQL Server database. This allows the mBLAST system to manage (input, edit, display, report) any kind of data without programming changes. The Data layer interacts with the Metadata layer.

The Data layer is stored in database tables in a Microsoft SQL Server database.

mBLAST System Technical Overview (Software Design Spec)

General Overview

mBLAST uses server-side ASP (Active Server Pages) and VBScript as the scripting language for delivering the HTML-based user interface. Client-side javascript is used minimally and only when necessary, to avoid problems with older browsers. Database access and business logic resides in COM objects that are written in Visual Basic 6. Microsoft SQL Server is as the SQL database, and stored procedures are used extensively to reduce traffic between the web server and the database server.

System Concepts

mBLAST is designed to allow arbitrary sets of data to be entered into the database, edited, and then published into HTML pages or XML documents. The information required to define the format and logic for inputting and outputting this data is stored in tables and can be edited by end users. The mBLAST system uses its own jargon for referring to the data and the definitions of the data, as follows.

Form: A form is a definition of an input screen. Included in the definition is the title of the form, and the name and type of each input field on the form.

Form Group: A form group is a set of related forms. When the user is entering data into a form, the system allows the user to navigate from one form to the next form within the form group.

Question: A "question" is actually the definition of an input field on the form. There are a number of input types available, including all of the standard html input types like text and textarea, as well as several enhanced types, like type for "customer", "phone number", or "date". Each type has a number of attributes that define how the question will appear on the page, and may include attributes that are specific to the question type. For example, the textarea type allows "rows" and "cols" to be defined, while the phone number type has an attribute that determine whether or not an extension is allowed.

Form Editor: The form editor is the user interface for defining forms, form groups, and questions. It's also known as "Edit Form Groups" within the mBLAST menu.

Form Generator: The form generator creates HTML files for each of the forms in a specified form group. The HTML files consist of a wrapper file (which contains include files for the template that the form is displayed within, and also includes the form itself), a form include file (which contains subroutine calls to the form engine for each of the questions on the form), a help file for the form, and two files that allow the data in the form to be displayed. The wrapper file allows the form to be added or edited, while the display files allow the form to be view.

Form Engine: The form engine is a set of subroutines in an include file (FormIncludes.asp) that display the various input types for the form, and allow the data on the form to be validated and saved.

Session: A session is an instance of data for the form. For example, for a company profile form, there will be a different session for each company that is entered into the form.

Core data: "Core data" refers to the data that is saved for a session. Because a form can allow any kind of data to be entered, the data is stored in generic files. So data from a company profile form is stored in the same generic "core data" files as data for a user profile. Each record in a core data file contains the value of a single question, for a single session. So if a form has twenty questions, there will be twenty records in core data for a session for that form. There are three core data files: core_int, core_varchar, and core_text. Core_int stores numeric data; core_varchar stores textual data up to 255 characters, and core_text stores textual data of more than 255 characters. Note that because each question is stored in a separate record, it is complicated to retrieve all of the questions for a form in a normal recordset format, where each record would correspond to a session, and each question to a field in the record. mBLAST uses stored procedures to convert core data into recordset formats.

Dataset: A dataset defines a set or group of sessions.

Presentation: A presentation is a definition of an output screen or screens. Included in the definition is pages that will be displayed; the sections within each the HTML for the page top, for between sections, and for the page bottom; indexes; and the data to put into the rows and columns in each section. Style sheets can be defined and elements of the style sheet can be applied to the elements of the page.

mBLAST's core data can be thought of as a database (a database with the SQL server database). A form or form group is a like a table. A question is like a field in the table. A session is like a record in the table. A dataset is like a view. This allows mBLAST to define new tables without needing to create new tables in the database, and also to create the HTML for entering, editing, and displaying the data from the tables.

Alert: When certain events on the system occur, an alert is raised. For example, whenever a form is saved, an alert is raised. The system allows actions to be defined for the alert. For example, someone can be notified, via email or IM, based on the message defined by the alert.

User Interface Concepts

After logging in the user, the system displays a "workspace UI". The UI is divided into several areas: the title bar, the workspace menu, the navigation bar, and the content area.

The title bar is simply a graphic that displays at the top of the screen.

The workspace menu is a set of buttons on the left side of the screen. Each button points to a module, or functional area, within the system. The set of buttons varies depending on the type of user. For example, a Marketing user sees a different set of buttons than a Publisher sees. In addition, the buttons may be grayed out if the user does not have sufficient rights to access the module.

The navigation bar displays under the title bar along the top of the screen. Included in the title bar is a drop-down sub-menu, which allows the user to navigate within a module. This sub-menu is populated based on an array that is set in each in ASP page. Other navigational elements may also appear within the navigation bar, including a link to context-sensitive help, and an Alias sub-menu, if the user has permission to access more than one company. Note that the navigation bar is used for providing access to the previous page or pages, rather than buttons, icons, or links on the form itself.

The content area is a white box in the middle of the screen where the main content for a page is displayed.

Within the content area, there are normally several elements: a module title, a page title, a page description, and a list or an input form. Each of the elements uses classes from the main style sheet to ensure that the look and feel is consistent from page to page. (Note: the style sheet classes were added only a few months back, so older modules in the system may not yet have been modified to use them).

The module title should correspond to the name of the module and remains the same on every page within the module.

The page title appears underneath the module title, on every page except for the module's main page (which doesn't need a page title). The page title is preceded by an mBLAST Logo bullet.

The page description is a short set of instructions for the user, which appears under the page title, and is optional. This area is also used to display error messages. Error messages always use a red font.

The list is an HTML table containing a list of records and action icons. The general format of the list is a title (describing the contents of the list); a "new" row for adding a new record, if applicable, one or more "edit" rows for viewing, editing, and deleting records; and a legend that describes each of the icons. Columns within the list generally include an ID column identifying the specific record, one or more columns containing information from the record (names, descriptions, etc.), and an actions column containing icons. The action icons allow the user to perform functions for the selected row, like add/edit/delete/view.

The input form is an HTML table containing an input form. The general format of an input table is a title (describing the contents of the form); a set of rows for each field in the form, containing a prompt on the left, and an input field on the right; and a Submit button at the bottom.

So a list is used to display a set of records with action icons, while an input form is used when adding, editing, or deleting a single record.

In many cases, the information that is being entered is more complicated than a single set of records. For example, in the form editor, the first page displays a list of the form group records, with add/edit/delete capabilities. But for each form group, there can be multiple forms, and for each form, there can be multiple questions. The UI handles this by adding a "View" icon for the set of child records. So in the form group list, there is an icon to "view forms" for the form group. Clicking that icon takes the user to another page that displays the form group that was selected, and that shows a list of the forms for the form group, with action icons for each form. On of the action icons for the form is to "view questions" for the form. This takes the user to another page, that lists the form group and form that was selected, and that shows a list of the questions for the form. And so on. So there is a hierarchical relationship between the parent and children forms. As the user navigates deeper into the hierarchy, each of the previous pages is added to the navigation bar's drop down list so that the user can navigate back to any of the previous pages.

Security

Each user has a user ID and password, which is validated against the Users table. The user can belong to one or more companies based on the User_Join_Company table. The User_Join_Company table contains a "Default" bit field for the company that the user is currently associated with.

When the user logs in, he is issued a SessionGUID, which is stored into a cookie. Each page on the site checks for this SessionGUID to determine who is logged in, and then checks to see whether the user has permissions to access the page. Permissions are granted based on whether the user belongs to a group that has access to the object that the user is trying to access. If the user does not have permission to access a page, he is redirected to an "access denied" page.

The SessionGUID also has an inactivity timeout associated with it. When the session times out, the user is redirected to the login page.

There are two tables in the system for handling object permissions: Objects and Object_Permissions. The Objects table defines each of the security objects. There are several types of objects. Type 5 is for menu items (or modules), while other types grant access to forms, hierarchies, questions, etc. The Object_Permissions define which groups have access to which objects.

Code Structure

Each ASP page in mBLAST is structured somewhat similarly, using a wrapper file and two code files. The wrapper file is stored in the /workspace directory (or a subdirectory of that directory), while the code files are stored in the /code/workspace directory (or a subdirectory of that directory).

An example of a wrapper file:

```
<%@ Language=VBScript %>
<%
strObjectId = 39
strObjectType = 5
strTitle = "mBLAST - mBLAST Internal"

CheckSecurity
```

```
%>
<!--#include virtual="/code/workspace/alert/company/alertlistupdate.asp"-->
<%
DisplayTemplateTop strTitle, True, True
%>
<!--#include virtual="/code/workspace/alert/company/alertlist.asp"-->
<%
DisplayTemplateBottom
%>
<!--#include virtual="/template/template.asp"-->
```

Note that Option Explicit is not used. While this would be preferable for performance reasons, it was not required when the system was originally designed and there has not been time to edit all of the pages to support it.

The strObjectId is the object used for permissions checking, and the strObjectType id the type of object (where 5 is a menu item or module). The value of strObjectId comes from the PageId field in the WorkspacePages table.

strTitle sets the HTML page title, and should be the same for every page in the module.

CheckSecurity checks to see if the user is logged in, if the SessionGUID is current, and if the user has permissions to access the page. If not, the user is redirected appropriately.

The first include file is an "update" page, which contains code that must be executed before any of the page's HTML has been written. This is used to handle form submissions (when editing a record for example), to retrieve recordsets and other data, and to set up the navigation submenu. Note that forms almost always submit to themselves rather than to another form. By convention this page is that same as the name of the wrapper page and code page, but with "update" appended.

DisplayTemplateTop writes the HTML at the top of the page, including the navigation bar and the menu...everything up to the content area of the page.

The second include file writes the HTML within the content area.

DisplayTemplateBottom writes the HTML at the bottom of the page...everything after the content area of the page.

The last include file is for the template subroutines, including CheckSecurity, DisplayTemplateTop, and DisplayTemplateBottom.

The wrapper file is used so that the page can be branded with a different look and feel. To do this, the wrapper file is copied to the branded directory, and the template include is changed to point to the branded template subroutines. This allows a single set of code pages to be used.

An example of an update file:

```
<%
Dim objAlert, objAlertRS

' Get alerts
Set objAlert = Server.CreateObject("mBAlert.CAlert")
objAlert.Active = 1
Set objAlertRS = objAlert.FetchAlertsByActive
Set objAlert = Nothing
```

```
' Set up jump array
Dim aryJump(1,0)
aryJump(0,0) = "/workspace/security/company/security.asp"
aryJump(1,0) = "Security Administration"

%>
```

Note that even though Option Explicit is not used, we still attempt to Dim all variables. This example shows a recordset being obtained by calling a method in a component. Then the "jump" array is set up. This array is used to define the drop-down list or sub-menu in the navigation bar.

A simplified example of a code file:

```
<%
Response.Write "<table border=0 cellpadding=1 valign=top>"
Response.Write "<tr><td><a class=sectionHeader>Security Administration</a></td></tr>"
Response.Write "<tr><td><a class=subheadermaroon><img
src=/images/bullet.jpg>&nbsp;Edit Alerts</a></td></tr>"
Response.Write "<tr><td>Use this page to view the alerts that are available, and to add
recipients for the alerts.</td></tr>"
Response.Write "<tr><td><font color=red>" & strError & "&nbsp;</font></b></td></tr>"
Response.Write "</table>"

Response.Write "<table class=contenttable cellpadding=3 cellspacing=1 border=0>"
Response.Write "<tr class=contenttitle><td colspan=3 class=contenttitle>Alerts</td></tr>"
Response.Write "<tr class=contentheader>"
Response.Write "<td class=contentheader>Id</td>"
Response.Write "<td class=contentheader>Description</td>"
Response.Write "<td class=contentheader>Actions</td>"
Response.Write "</tr>"

If Not (objAlertRS Is Nothing) Then
 Do Until objAlertRS.EOF
  Response.Write "<tr class=contentedit>"
  Response.Write "<td class=contentedit>" & objAlertRS.Fields("AlertId") & "</td>"
  Response.Write "<td class=contentedit>" & objAlertRS.Fields("Description") & "</td>"
  Response.Write "<td class=contentedit><a href=recipientlist.asp?AlertId=" &
objAlertRS.Fields("AlertId") & "&PrimaryMessageId=" & objAlertRS.Fields("MessageId")
& "><img border=0 height=16 width=16 src=/images/hier_pers.gif alt=""Edit
recipients""></a>"
  Response.Write "</td>"
  Response.Write "</tr>"
  objAlertRS.MoveNext
 Loop
 Set objAlertRS = Nothing
End If

Response.Write "<tr><td colspan=3>"

Dim aryLegend(0,1)
aryLegend(0,0) = "edit recipients"
aryLegend(0,1) = "hier_pers"
DisplayLegend aryLegend, 2

Response.Write "</td></tr>"
Response.Write "</table>"

%>
```

This illustrates the use of classes to define the various elements of the page (module title, page title, page description, list form including title, edit row, and legend.)

Also, note that Response.Write is used to output all of the HTML, rather than context switching between VB code and HTML. Eliminating the context switching allows the code to execute slightly faster in the current versions of ASP.

Components

Perhaps the most important component is mBData. This component is used to access the SQL database. It should be used for any database access, both from within a component and from within an ASP script. (Most of the time, you should not need to access mBData directly from ASP, since there should be another component that you can use).

mBData allows both SQL calls and stored procedure calls. However, use of SQL calls is deprecated and will cause your code to be rejected during our review process. To access the database, you will first need to write a stored procedure, then access that stored procedure via mBData. mBData supplies two methods for stored procedures: ExecuteSP and ReturnSPResults. The first simply executes the procedure, while the second returns a record set from the stored procedure. Neither method supports the use of output parameters, so the stored procedure should be written to return a result set.

Here's an example of a call to mBData, from within a component:

```
    Set objData = CreateObject("mBData.CData")

    Set objRS = objData.ReturnSPResults("InsertCompany", _
    adVarChar, 255, pstrCoName, _
    adInteger, 4, pintPricingLevelId, _
    adInteger, 4, pintCoType)

  Set objData = Nothing
```

Note that you specify the stored procedure's parameters by using the parameter type (adVarChar), the length of the data (255), and the data (pstrCoName).

The following briefly describes the purpose of mBLAST components. Components that are obsolete or are not actively being used are not listed. We can provide source code for individual components as needed for your project.

mBAlert.dll -- Methods for adding, editing, deleting, and fetching alerts, and for raising alerts.

mBAreaInterest.dll -- Methods for adding, editing, deleting, and fetching "Areas Of Interest" and scheduling alerts for areas of interest (where the user specifies his interests using system hierarchies).

mBAuthor.dll -- Methods for adding, editing, deleting, and fetching form groups and related tables like Questions.

mBBulkData.dll -- Methods for importing and exporting data in XML format.

mBBulkFeed.dll -- Methods for adding, editing, deleting, and fetching bulk feeds (schedules of data feeds to remote systems).

Introduction

This document is indended to serve as the working Application Design Specification for the mBLAST project, and will include both high and low level design specifications.

mBLAST Overview

At its core, mBLAST is a massive database of product/service data, company and personnel information, contact information, personal preference information, event information, messages and communications, news, images and just about every other conceivable type of digital marketing collateral.

The mBLAST Gateway is a rich supporting layer of web-based applications for inputting, accessing, comparing, sharing, challenging, hosting and researching this data. MBLAST also offers a variety of supporting applications for the exchange of data such as bulk data transfer, email notifications, and hosted environments.

A key to the power of mBLAST is that every unit of data accumulated will allow a wide variety of information to be attached to it. A supporting data layer will contain meta and supporting data about each product/service/company data field, and will allow for expansion of the supporting data as the scope of the application grows.

System Overview

The architecture of mBLAST uses a familiar n-tier web-application paradigm; The back-end network may consist of data warehouses, messaging servers, transaction servers, application servers and other supporting components connected to the internet. The middle-tier application layer will consist of component applications that communicate with the back end layer and the user interface. The user interface will consist of, for the most part, web pages hosted on a public web server and dynamically generated from script-based templates. This architecture allows for maximum flexibility and scalability. Any piece, or entire layer, can be replaced or modified at any time without compromising existing functionality.

We have chosen a Microsoft-based platform for the initial release of mBLAST Gateway. This platform is robust, and offers a variety of tools that can rapidly increase the time to market for web-based applications. However the mBLAST architecture is not dependent on the Microsoft platform, which allows the opportunity to migrate some or all of the application to another platform at any time.

Design Considerations

The following are issues that need to be considered when determining a design and implementation approach for the mBLAST application.

Related software or hardware

The application should be designed for cross-platform portability. This does not mean that all application components must be cross-platform capable from day one, but the application architecture must be such that each tier can be ported to another platform without affecting the performance of any other tier.

End-user characteristics

The application should be designed with maximum flexibility in mind, both in terms of functionality and presentation, as the scope of the application will continually evolve and change, and the design will need to support these changes easily and without requiring redesign efforts.

General Constraints

The following are constraints that have a significant impact on the design:

- Database Size – Because we anticipate a very high volume of data, potentiall terabytes of data, the Data Layer will need to be designed accordingly.
- Database Performance – Because we anticipate a high volume of users, and also plan on implementing automated processes that will further increase the dB Server usage, the Data Layer will need to be designed accordingly.

Goals and Guidelines

Following are priorities that dominate the design of the system's software:

- Maximum Flexibility in UI modifications
- Ability to easily accept new data categories and types
- Bring Core functionality to market first, which allow for the collection of data, then focus on output capability

Architectural Strategies

Summary

The project should be designed using an N-tier approach. Specifically, the program should separate the User Interface, Business Logic, and Database Access. Any of the tiers should be able to be modified without any adverse effects, as long as the interface between the tiers is not changed.

The UI will be HTML output to be displayed by browsers. The UI is any type of server side scripting language. The Business Logic consists of components either running on the web server or a separate transaction server and will be responsible for calling the Database Access components, manipulating the data according to predefined business rules, and passing this information to the UI. The Database Access will be components running on the web server or a separate transaction server and will be responsible for any interaction with the database.

UI Tier

For this project, the UI will consist of Active Server Pages. For scalability reasons, the ASP should not be dependent on any session variables. This will enable the application to be run on a web farm where it is unlikely that the same server will be used for all requests. Using WLBS (Windows Load Balancing Software), it's possible to cluster up to 32 servers in a single web farm for scalability and/or redundancy.

For security reasons, no sensitive information such as usernames and passwords should be contained in the ASP. Instead, this information should be stored in another location such as the registry or UDL files.

Custom Written Components

All custom written components should first undergo a buy vs. build analysis. If it is determined that custom components need to be written, they should adhere to the following guidelines:

- All components should be named MBComponentName to make them easily identifiable.
- All components should be written to run on MTS (Microsoft Transaction Server).
- All components should be written in VB (Visual Basic) unless otherwise specified.
- All components should have an API documenting its exposed Properties, Methods, Events and dependencies on other components.
- All components should write all unrecoverable errors to an error log.
- At a minimum, all components should consist of the following properties:

  MsgLogPath – The complete path to the log file to be used for writing unrecoverable errors.

  IniPath – The complete path to the INI file used.

  NodeName – The name of the computer where the component is running.

Business Logic Tier

The components that make up the business logic should not have any direct interaction with the database. All information residing in a database should be accessed using the Data Access Tier.

Data Access Tier

The Data Access Tier should be written using ADO for database access. Whenever possible, stored procedures should be used instead of embedded SQL. This provide better performance and should make the transition from SQL Server to Oracle or another highly scalable RDBMS easier.

System Architecture

This section describes the main areas of application functionality.

mBLAST Gateway Environments

The mBLAST Gateway is the web-based user interface portion of mBLAST . The Gateway is accessible via the World Wide Web, though registration is required for access by end-users. The Gateway consists of three main areas; data input, data output and data management. The Gateway provides a comprehensive collection of applications for performing these functions. The data management capabilities are provided in part by "My mBLAST ", a customizable end-user environment for leveraging the features of mBLAST .

Data Access

The Access Modules of the program allow for output clients to receive the data from the gateway. In the On Demand mode, users enter the system via a USERID/PASSWORD entry. They can view many different listings of the data, depending on their particular interest. Since the data is hierarchical (industry, product group, company, product line) users can click down to as discrete a listing as they require. Information is viewable in a listing format. There is also a matrix view format that allows users to either Select All or Select Specific Sellers to create a custom matrix; users have the option of selecting all possible fields for that category or selected fields as well. There is a transpose option that allows views with either features or Sellers across the top horizontal headers, with the Sellers or features in the vertical headers. Users can save data viewed as a file and send it as an email attachment .Data output modules include Buyer's Guide Matrix, Side-by-Side Comparison Engine, Company Profiles, Link Pages, Data Mining Engine, Glossary, etc.

COMPONENTS

- Matrix Generator – accept user input to generate custom matrixes of mBLAST data – ability to save Matrix designs in user account
- Company Profiles –
- Link Pages -
- Data Mining Engine – sophisticated search capabilities across mBLAST data sets
- Glossary -
- Data Packaging– Generate downloadable/email attachment file from screen data

- Data Validation– allows a variety of supporting data to be attached to each data unit such as challenges and comments.
- Data Spider– configurable search agent that spiders internet sources for pertinent data
- Data Object – Coordinates database queries (searches, data retrieval, add/delete/update records)
- Messaging Engine

Data Input

In this module, corporations can enter their data into predefined fields. Users enter the system via a USERID and PASSWORD-protected entry. Emphasis of the interface is on click boxes and radio buttons, with as little open text entry as possible to maximize the apples-to-apples comparison of the data. Each company/division has its own profile in the database core against which all data is attached. Corporations can select what industries and product categories in which they participate, and for each product category they are presented with additional screens for data entry, with most products having about 25 fields on average representing that product category.

COMPONENTS

- Form Generator – creates web-based forms based on pre-specified criteria (using the Administrator Form Creator)
- Data Query Module – Coordinates database queries (searches, data retrieval, add/delete/update records)

mBLAST Workspace

Customizeable environment for registered mBLAST users. Provides easy access to mBLAST features and personalized services, including messaging, “delta” data change notifications, automated data collection engine, etc.

COMPONENTS

- Matrix Generator – accept user input to generate custom matrixes of mBLAST data – ability to save Matrix designs in user account
- Company Profiles –
- Link Pages -

- Data Mining Engine – sophisticated search capabilities across mBLAST data sets
- Glossary -
- Data Packaging– Generate downloadable/email attachment file from screen data
- Data Validation– allows a variety of supporting data to be attached to each data unit such as challenges and comments.
- Data Spider– configurable search agent that spiders internet sources for pertinent data
- Data Object – Coordinates database queries (searches, data retrieval, add/delete/update records)
- Messaging Engine

mBLAST Gateway Supporting Applications

In addition to the central functions of inputting and extracting data, mBLAST Gateway offers supporting applications to make working within the mBLAST environment easier and

Automated Supplemental Data Collection Engine

Several other tools will be offered to the input and output side clients that help create a set of capabilities that builds repeat visits and loyalty to the gateway itself. One of these is the Supplemental Data Collection Engine, which is an automated robot/spider that is constantly searching the Web for information pertinent to different tools on the site. For instance, PR personnel are likely to be interested in how their latest press release was covered by various publications, newsgroups, Web sites, etc. Specific press releases could be tied to a data collection spider that would visit regular sources of content for their coverage of the topic, driven by key words from the release. This engine could be used to drive competitive tracking capability for companies, or competitive article tracking tools for reporters who want to see how others covered their same story. This tool will be accessible throughout the gateway in different ways.

COMPONENTS

- Data Spider Component – configurable search agent that spiders internet sources for pertinent data

Messaging Engine

The Messaging Engine is a powerful, email-based tool present throughout the gateway's screens. On different screens it provides different support. For instance, in the Data Access On Demand Module/Matrix Mode, it enables the user to send a copy of the matrix shown to all public relations managers for each of the companies listed – so a reporter getting ready to run a story based on those facts can shoot the matrix off to the pertinent PR personnel for a final lookover before publication. In the Data Access On Demand Module/Listing Mode, it enables the user to send an email to all the PR personnel for the companies listed/checked, replacing many individual calls or emails to accomplish the same thing. Indeed, as the PR personnel register their names and areas of responsibility on the input side of the gateway, they enable a communications path to be created on the output side that makes the overall communications process far more efficient and direct. Future plans include enhancements to include functionality associated with instant messaging and unified messaging.

COMPONENTS

- Messaging Engine

Data Validation Engine

A sophisticated data validation module enables clients to double check data, to make use of the email engine to send data around the client's firm, and to perform side-by-side comparisons of products against the competition. In essence, this is a valuable competitive analysis and product management engine for marketing departments and product managers. Companies can enter proposed products online in a draft mode to see how they would stack up with current competitive products. Information is presented in an easy-to-understand matrix format. Companies can also enter data in advance and embargo the data so that it is only available to the public at a specific point in the future – such as would be required with a new press released product.

A key part of the data validation module – and where it draws its name – is the comment communications chain capability, which is visually demonstrated by the "Challenge" button. The Challenge button is an icon in the upper, right-hand corner of the screen that allows the present corporate marketing user to contest information presented by a competitor. So, if

Company A believes that a cell of information from Company B in the presented matrix is not quite true, they can click on the Challenge icon and then click on the cell in question, and an interactive comment field is created. Company A would enter the rationale for questioning the data (the questioner, as initially planned, would be anonymous to Company B but not to mBLAST) and hit submit. An email is sent to 1) the designated PR contact for that product at Company B, and 2) mBLAST 's staff, for monitoring. The Challenge icon remains in the corner of the cell in question until a response from Company B is received by the gateway. Upon receipt of a response (notification of which is sent to 1) the questioner at Company A and 2) mBLAST 's staff for monitoring), the icon reverts to a Comments Available icon which permanently remains associated with that cell.

This feature is critical to the success of the mBLAST business model as it provides a mechanism for those who know most about those products – the Sellers themselves – to keep each other in check. Since this data is sent out to many different output clients, it becomes a critical part of corporate positioning to maintain watch over competitor data as well as one's own. It also relieves mBLAST of the burden of having to play middleman. Note that mBLAST is involved in the early stages of this process in a monitoring capacity; this is to watch for abuse and misuse of the feature. mBLAST may have to limit the number of times that this feature may be invoked to prevent some companies from questioning every single cell of every single competitor. (There is more discussion of this feature under Data Collection and Management heading later in this document.)

Importantly, these comment streams also provide Advisors with great angles for potential stories. In this sense, corporate positioning can be applied on top of mere data in this data stream.

COMPONENTS

- Data Validation Module – allows a variety of supporting data to be attached to each data unit such as challenges and comments.

mBLAST Support Applications and Value-Added Services

mBLAST also offers a variety of support applications and value-added services to clients and end-users, including:

Data Entry-Automated Push/Pull Module

If a firm already has product and corporate information databases, mBLAST 's gateway has open APIs that allow direct interface to the site for continual updating. mBLAST will accept any data that conforms to its data structure – using a record layout format publicly available on the gateway. mBLAST will accept data files in both tab delimited formats or XML coded formats. Direct, real-time feeds are also possible for select applications where it makes sense, such as with service availability (i.e., where service is presently available; this data tends to be highly proprietary, but makes sense to funnel through a gateway for ease of use). In this sense, the gateway might be "pulling" data from the client sites as opposed to them pushing data to the gateway.

Data Access-Automated Push/Pull Module

The direct push/pull module is more tuned towards self-feeding APIs. An output client might regularly download a new file to incorporate on its site or might choose to receive alerts from the gateway with "delta" records of changed data. Data is sent out in a specific format, and output clients are expected to create/adjust their programming to take advantage of these feeds. For the largest shopping portals such as AltaVista and Yahoo!, mBLAST will conform to their APIs for automated distribution of product information to these sites on behalf of its selling organizations.

Hosting

mBLAST will offer ASP-like hosting services, which will allow Web sites to include Buyer's Guides and other mBLAST modules in their Web sites without incurring the expense of site programming fees. mBLAST will work with customers to adopt the look and feel of their sites on its servers, providing a seamless experience for customers. mBLAST intends to resell the hosting services of its partners in providing these services.

Subsystem Architecture

This section further describes components listed in System Architecture above.

Product/Industry Creator

The Product/Industry Creator component allows the addition of new product/service/industry types into the Product/Service database. This component also allows the creation of relationships between the new category and any available hierarchies.

Functionality:

1. Admin identifies new product/service/industry
2. Admin selects available hierarchies and identifies the correct position or coding sequence for the new category in each hierarchy

Form Creator

The Form Creator component is a foundational piece of the mBLAST solution. The Form Creator has several functions:

- Allow an administrator to create or modify an input form using a web-based interface
- Allow the admin to attach the new input form to a specific product category or industry or modify the product category for an existing form.
- As the new form is created, add the necessary information to the Fields table, the Fields_Join_Product/Service table, Fields_Join_Form table and the Form table to allow for the dynamic generation of forms and the correct insertion of data from the form into the Core Data tables and the supporting tables.
- As a form is created or modified, generate and save the input form as an ASP page and create a reference to the location of the file in the Form Pages table.

Functionality:

1. Administrator initiates a new form
2. Admin identifies the product category for the new form
3. Admin identifies the previous form in the form chain (if any)
4. Admin identifies form questions (i.e. Product Name, Product Color etc) and creates the specific input form type for each question
5. Form creator adds the input form configuration information to the Forms table (i.e. Question, input type, validation type, length, order on page, etc)
6. Form Creator ads form questions to the Fields table as the field names for the new product type and maps them to columns in the Core Data tables (i.e Product Name, Column #1, Core_Data_Varchar)
7. Creates relationships in Form_Join_Fields table between the Form and Fields tables
8. Generate the ASP form page and add the form page location to the Form_Pages table.

Core Data Object

This component makes modifications to the Core Data tables and supporting tables based on information from an input form. This component can retrieve (load) data from the Core Data, insert records into the Core Data, update records in Core Data, and delete records from Core Date. This component is also responsible for updating the supporting Join tables, and other supporting tables such as the History table, Product/Service table, etc. This object will perform most of the data manipulation required by other objects. This Object will communicate with a set of stored procedures and pass the required values to allow the stored procedures to perform the required task.

Matrix Creator/Generator

The Matrix Creator/Generator component allows an end-user to create a table of product/service information. The user will be provided with an administrative screen to select the products and product features to include in the matrix. Once the user has selected the matrix parameters, a matrix/table will be generated using Core Data. The user will have the ability to save this configuration for viewing at a later date.

Functionality:

1. The user initiates a new matrix for a particular product type.
2. A matrix creation screen is created with a matrix identifying all available products and all available product features. The user selects the products and features to include in the matrix
3. A new record is added to the Matrix table identifying the Matrix name, and relationships are created in the Matrix_Join_User table, Matrix_Join_Fields table, and Matrix_Join_Products table.
4. The Matrix is generated using the new relationships stored in the database.
5. If the User decides to save the matrix, the data is maintained, otherwise all new matrix data is deleted from database.

Data Packager

The Data Packager component allows the creation of a downloadable or email-attachable file from any data output. For example, a user creates a Matrix, and wants to publish it on her website. Using the Data Packager, she can create a downloadable copy of the matrix in HTML format with all supporting images included. The Data Packager will generate the file and initiate the download. The Data Packager should be able to create multiple file types such as HTML, Excel spreadsheets, Delimited Text File, Access Database, etc.

Data Qualifier

The Data Qualifier component is a key feature of the mBLAST Gateway. The Qualifier allows an infinite amount of supporting or meta information to be attached to the Core Data. The meta information facilitated by the Qualifier is another data type that will actually reside in the Core Data tables, and relationships between the supporting data and the Core Data will be provided by the Core_Join_Meta table. Another table called Meta will contain the list of Qualifier data types such as challenge, response, positioning statement, etc

An example of this "qualification" ability of the Gateway is allowing users to challenge the claims made by competitors, and allowing the competitor to respond to this challenge. This exchange will be permanently linked to the actual data value in question, and will be a viewable part of the data value history.

Image Uploader

The Image Uploader Component allows input users to upload images of various file types. These images can be for Company logos, employee portraits, or product shots. The Image Uploader will allow the user to browser their hard drive, select a file, upload the file to a temporary directory, verify the file type, rename the file using a standard naming convention, and move the file to the correct image directory. The Uploader will also add the file path and name to the corresponding Product/Service/Company/Employee Core Data record.

Automated Delta Data Notifier

The Delta Data Notifier allows users to be automatically notified in the event that any selected Core Data is modified. A change to the Audit table should trigger a search of all users that have chosen to be modified regarding that data unit. Leveraging the messaging component, the pertinent users will be notified via email, pager, etc. or a notice will be included in their mBLAST Workspace environment. For each flagged data unit, a record will be included in the Delta_Notify table and a relationship to the user will be created with the Delta_Join_User table.

Messaging Component

Data Spider Component

Configurable search agent that spiders Internet sources for pertinent data

Automated Input Push Component

API connections to mBLAST data core for automated push delivery of data by input clients

Automated Input Pull Component

API connections allowing mBLAST automated retrieval of data from input client databases

Automated Access Push Component

API connections to output client databases allowing automated push delivery of mBLAST content

Automated Access Pull Component

API connections to mBLAST data core for automated retrieval of data by output clients

The hosting service will leverage the User Interface Data Access Objects utilized by the mBLAST Gateway. An additional object layer will coordinate the administrative aspects of the hosting solution.

Hosting configuration Component

Branding Component

Coordinates the branding of the hosted solution to conform to the client site graphics/navigation etc.

Form Creator

The Form Creator component is a foundational piece of the mBLAST solution. The Form Creator has several functions:

Allow an administrator to create or modify an input form using a web-based interface

Allow the admin to attach the new input form to a specific product category or industry or modify the product category for an existing form.

As the new form is created, add the necessary information to the Fields table, the Fields_Join_Product/Service table, Fields_Join_Form table and the Form table to allow for the dynamic generation of forms and the correct insertion of data from the form into the Core Data tables and the supporting tables

As a form is created or modified, generate and save the input form as an ASP page and create a reference to the location of the file in the Form Pages table.

Step-by-step:

Administrator initiates a new form

Admin identifies the product category for the new form

Admin identifies the previous form in the form chain (if any)

Admin identifies form questions (i.e. Product Name, Product Color etc) and creates the specific input form type for each question

Form creator adds the input form configuration information to the Forms table (i.e. Question, input type, validation type, length, order on page, etc)

Form Creator ads form questions to the Fields table as the field names for the new product type and maps them to columns in the Core Data tables (i.e Product Name, Column #1, Core_Data_Varchar)

Creates relationships in Form_Join_Fields table between the Form and Fields tables

Generate the ASP form page and add the form page location to the Form_Pages table.

Core Data Object

This component makes modifications to the Core Data tables and supporting tables based on information from an input form. This component can retrieve (load) data from the Core Date, insert records into the Core Data, update records in Core Data, and delete records from Core Date. This component is also responsible for updating the supporting Join tables, and other supporting tables such as the History table, Product/Service table, etc. This object will perform most of the data manipulation required by other objects.

Matrix Creator/Generator

The Matrix Creator/Generator component allows an end-user to create a table of product/service information. The user will be provided with an administrative screen to select the products and product features to include in the matrix. Once the user has selected the matrix parameters, a matrix/table will be generated using Core Data. The user will have the ability to save this configuration for viewing at a later date.

Step-by-step:

The user initiates a new matrix for a particular product type.

A matrix creation screen is created with a matrix identifying all available products and all available product features. The user selects the products and features to include in the matrix

A new record is added to the Matrix table identifying the Matrix name, and relationships are created in the Matrix_Join_User table, Matrix_Join_Fields table, and Matrix_Join_Products table.

The Matrix is generated using the new relationships stored in the database.

If the User decides to save the matrix, the data is maintained, otherwise all new matrix data is deleted from database.

Data Packager

The Data Packager component allows the creation of a downloadable or email-attachable file from any data output. For example, a user creates a Matrix, and wants to publish it on her website. Using the Data Packager, she can create a downloadable copy of the matrix in HTML format with all supporting images included. The Data Packager will generate the file and initiate the download. The Data Packager should be able to create multiple file types such as HTML, Excel spreadsheets, Delimited Text File, Access Database, etc.

Data Qualifier

The Data Qualifier component is a key feature of the mBLAST Gateway. The Qaulifier allows an infinite amount of supporting or meta information to be attached to the Core Data. The meta information facilitated by the Qualifier is another data type that will actually reside in the Core Data tables, and relationships between the supporting data and the Core Data will be provided by the Core_Join_Meta table. Another table called Meta will contain the list of Qualifier data types such as challenge, response, positioning statement, etc

An example of this "qualification" ability of the Gateway is allowing users to challenge the claims made by competitors, and allowing the competitor to respond to this challenge. This exchange will be permanently linked to the actual data value in question, and will be a viewable part of the data value history.

Image Uploader

The Image Uploader Component allows input users to upload images of various file types. These images can be for Company logos, employee portraits, or product shots. The Image Uploader will allow the user to browser their hard drive, select a file, upload the file to a temporary directory, verify the file type, rename the file using a standard naming convention, and move the file to the correct image directory. The Uploader will also add the file path and name to the corresponding Product/Service/Company/Employee Core Data record.

Automated Delta Data Notifier

The Delta Data Notifier allows users to be automatically notified in the event any selected Core Data is modified. A change to the Audit table should trigger a search of all users that have chosen to be modified regarding that data unit. Leveraging the messaging component, the pertinent users will be notified via email, pager, etc. or a notice will be included in their mBLAST Workspace environment. For each flagged data unit, a record will be included in the Delta_Notify table and a relationship to the user will be created with the Delta_Join_User table.

Still In Production

Messaging Component

Data Spider Module

Automated Input Push Module – API connections to mBLAST data core for automated push delivery of data by input clients

Automated Input Pull Module – API connections allowing mBLAST automated retrieval of data from input client databases

Data Spider Module – configurable search agent that spiders internet sources for pertinent data

Automated Access Push Module – API connections to output client databases allowing automated push delivery of mBLAST content

Automated Access Pull Module – API connections to mBLAST data core for automated retrieval of data by output clients

Hosting

The hosting service will leverage the User Interface Data Access Objects utilized by the mBLAST Gateway. An additional object layer will coordinate the administrative aspects of the hosting solution.

Hosting Administrative objects:

Hosting configuration module – could be coded into templates

Branding module – coordinates the branding of the hosted solution to conform to the client site graphics/navigation etc.

mBLAST Application Scenarios

Create/Edit/View Product Categories (mBLAST.org)

Product categories can be defined from multiple sources: some industries have standard product definition codes, such as within the UN/SPSC system, but others, like DSL, do not yet have adequate representation in standardized categorization schemas, so they rely on other product category definitions. Two sources of such definitions, are industry publications and the industry associations and forums. Others may be from RosettaNet and other similar organizations.

mBLAST seeks to have both standardized and non-standard schemas represented. If standardized mappings exist between schemas, mBLAST will adopt those mappings, however, mBLAST will not seek to create mappings between categorizations. Changes, additions, and deletions to the various product category definitions will be considered an alert event. E-mail alerts would be sent to all end users which have associated themselves with a product category. There would be a process to permit the end users to review and change their previous category selections. Changes made to the end users' product category selections, would also be considered an alert event, sending an e-mail alert to all end users having made a previous association with this end user through their product category definitions. Those users, primarily reporters/editors, would have the opportunity to make changes to their associations.

Create Product Categories

End users (Associations/Forums, Publishers) including mBLAST administration, need a process to create new product categorizations. The process for creating a new product categorization scheme is:

- Choose the author of the categorization from a list (Network World, DSL Forum, mBLAST)
- Choose the industry topic that the categorization belongs to (ATM - Banking, ATM - Networks, DSL, MSP) from an existing list of standard industries (SIC/NAICS)
- If no standard industry listing exists for that topic, permit a new industry topic category to be created (multiple industry associations must be permitted.)

- If a standard industry is chosen, and a standard product definition exists for that industry, list the likely choices of product category definitions. (Example: if Computers is the industry topic chosen, then there exists a RosettaNet definition for Computers. List only those DTD's that would be pertinent to the computer industry. Based on the choice made in the RosettaNet portion of the page, and if a standard mapping exists between the RosettaNet DTD and a UN/SPSC code, then the UN/SPSC portion of the page would have defaulted to the mapped value for that product/service code, and vice versa.)
- If no standard product definition exists, or the existing choices are inadequate, within RosettaNet and/or UN/SPSC, allow the end user to request information from the respective organizations via email.
- Allow end user to input multi-level product categories and sub-categories.
- Send email alert to associated entities to inform them of the new product category definitions.
- E-mail contains URL to a web page of new product categorizations.

Edit Product Categories

End users (Associations/Forums, Publishers) including mBLAST administration, need a process to view/edit existing product categorizations, and a method to add a new product category within an existing category. The process for viewing/editing an existing product categorization scheme is:

- Choose the editor of the categorization from a list (Network World, DSL Forum, mBLAST)
- Choose the industry topic that the categorization belongs to (ATM - Banking, ATM - Networks, DSL, MSP) from an existing list of standard industries (SIC/NAICS)
- View the existing product categorizations.
- Add/Edit product categorizations, including inserting a new subcategory within an existing category.
- Send email alert to associated entities to inform them of the changed product category definitions.

- E-mail contains URL to a web page of all changes, additions, and deletions to the product categorizations.

Selecting Product Categories

End users (Marketing, MarComm, PR Firms) select from as many categorization schemas as they want to be associated with, from all categories that mBLAST has represented for their industry or industries. The process for these users to select their product categories is:

- Select product categorizations from those available for the end user's industry associations as represented on input pages, and/or
- Select product categorizations from the URL sent when additions/changes are made (see above)
- Choose to be alerted when product category definitions are created/edited
- When product selection has changed, alert all associated entities.

End users (Reporters/Editors) select their "beats", or areas of interest, from the list of product categories in mBLAST. The process for these users to select their areas of interest is:

- Select areas of interest from those available (i.e., industry(s), product categories, companies) on input pages and/or
- Select product categorizations from the URL sent when additions/changes are made (see above)
- Choose to be alerted when changes to their areas of interest occur

Create/Edit Survey

Create a New Survey

A survey is a questionnaire that end users can create, and then invite companies to fill out. The process for creating a survey is:

- Give the survey a name
- Choose default questions for the survey
- Define additional questions for the survey
- Preview survey, allow edits, additions
- Invite companies to fill out the survey (survey cannot be changed after this point)

  Allow survey author the ability to choose whether survey respondents can see their answers and those of the other respondents
- Display the survey results in presentation formats
- Ask if this is a one time or repeat survey
  - One time surveys can be saved to the Edit Saved Surveys workspace for a limited time
  - Repeat surveys are saved in the Edit Saved Surveys area of the workspace

Edit a Saved Survey

Editing a saved survey gives the end user the ability to modify which questions are included and which questions are grouped together. The process for editing a survey is:

- Preview Chart from the Create New Survey process, or
- Choose a previously saved survey from Edit Saved Survey in the workspace, or
- Choose Edit Matrix from the chart view,
- Choose columns of data to be eliminated from the view, and/or choose columns of data to be grouped together.
- Preview survey, allow edits, additions
- Invite companies to fill out the survey (survey cannot be changed after this point)

  Allow survey author the ability to choose whether survey respondents can see their answers and those of the other respondents
- Display the survey results in presentation formats
- Ask if this is a one time or repeat survey
  - One time surveys can be saved to the Edit Saved Surveys workspace for a limited time
  - Repeat surveys are saved in the Edit Saved Surveys area of the workspace

Edit a System Survey

System surveys are predefined surveys and exist in the mBLAST system, by product category. End users can choose one of these surveys as a "starting point" in their creation process, using the matrix tools to edit the survey. The process for editing a system survey is:

- Drill down through a product category to see if a system survey exists
- Select existing survey or create new survey (if new go to create new survey process)
- Choose Edit Matrix from the chart view,
- Choose columns of data to be eliminated from the view, and/or choose columns of data to be grouped together.
- Preview survey, allow edits, additions
- Invite companies to fill out the survey (survey cannot be changed after this point)

  Allow survey author the ability to choose whether survey respondents can see their answers and those of the other respondents
- Display the survey results in presentation formats
- Ask if this is a one time or repeat survey
  - One time surveys can be saved to the Edit Saved Surveys workspace for a limited time
  - Repeat surveys are saved in the Edit Saved Surveys area of the workspace
- Save matrix with new name in the authors' workspace
-

Future Development: Vendor request will determine when a series of repeat questions becomes included in system surveys.

Create/Edit Data Presentation

Create a New Presentation

End users can create a data presentation format based on predefined templates. The process to create/edit a data presentation is:

- Create a chart based on a template, choices are:
    - Buyer's Guide, Company Profile, Links Pages, Directory, Glossary of Terms.
- Configure the presentation format from predefined configuration criteria
- Generate the chart

Edit a Saved Presentation

End users can choose to use a previously saved presentation format. The process to edit a previously defined format is:

- Select a previously saved presentation from the list
- Modify the chart by deleting and/or grouping columns

Edit a System Presentation

End users can choose to use a previously defined system presentation as a starting point for their own presentation. These presentations are listed by product category. The process to use a system presentation is:

- End user locates the desired product category and drills down to the data presentation which exists for that product.
- Modify the chart by choosing Edit Matrix tool; allows deleting/grouping columns.

Note: The basic steps of survey creation/editing and data presentation creation/editing are the same. The distinction between the two needs to be made for the end user.

Browse mBLAST

The browse mBLAST feature is a suite of tools to allow users to search mBLAST data in a variety of ways. There are four basic Browse types: View by Hierarchy, View by Topic, Search Database, and Search by Type.

View by Hierarchy

The View by Hierarchy feature allows the user to browse through a product/service hierarchy, and generate a matrix based on a product category. By clicking on a product category a matrix is automatically generated, and then the user can edit and save the matrix just like any other presentation.

View by Topic

The View by Topic allows the user to find companies by proprietary hierarchies. The View by Topic feature will display all available proprietary hierarchies, possibly by using the Core Hierarchy as a method to search. Once a proprietary hierarchy has been defined, the browse mechanism works in the same way as the View by Hierarchy feature (see above).

Search Database

The Search Database feature is a full-text search engine that can be directed at a variety of data types. The types include:

Corporation Profile
Publication Profile
Editor/Reporter Profile
Product/Service Data
Categorization Headings
Website Address
Logo

Executive Pictures
Direction to Entity
Glossary

Each of these data types searches on a specific subset of the Core Database, and uses one or more fields as the basis of the search. For example, the Corporate Profile Search would only search on Company records, and may use the Company Name, and Company Description fields as the basis of the search. The results are returned in standard search engine format.

Search by Type

The Search by Type feature allows the user to select a data type and then generate a screen display of a single record within the database. The data types available are the same as the Search Database feature (see above). When the user selects a Type, they are then presented with a method to select a specific company or product. The presentation format for that Type is then displayed with the requested data.

It should be obvious from the above-discussed apparatus embodiment that numerous other variations and modifications of the apparatus of this invention are possible, and such will readily occur to those skilled in the art. Accordingly, the scope of this invention is not to be limited to the embodiment disclosed, but is to include any such embodiments as may be encompassed within the scope of the claims appended hereto.

I Claim:

A method for collecting marketing information collateral data in all electronic forms, said method comprising the steps of:

inputting the data;

managing the content of the data;

editing the data; and

controlling the dissemination of the data to the public.

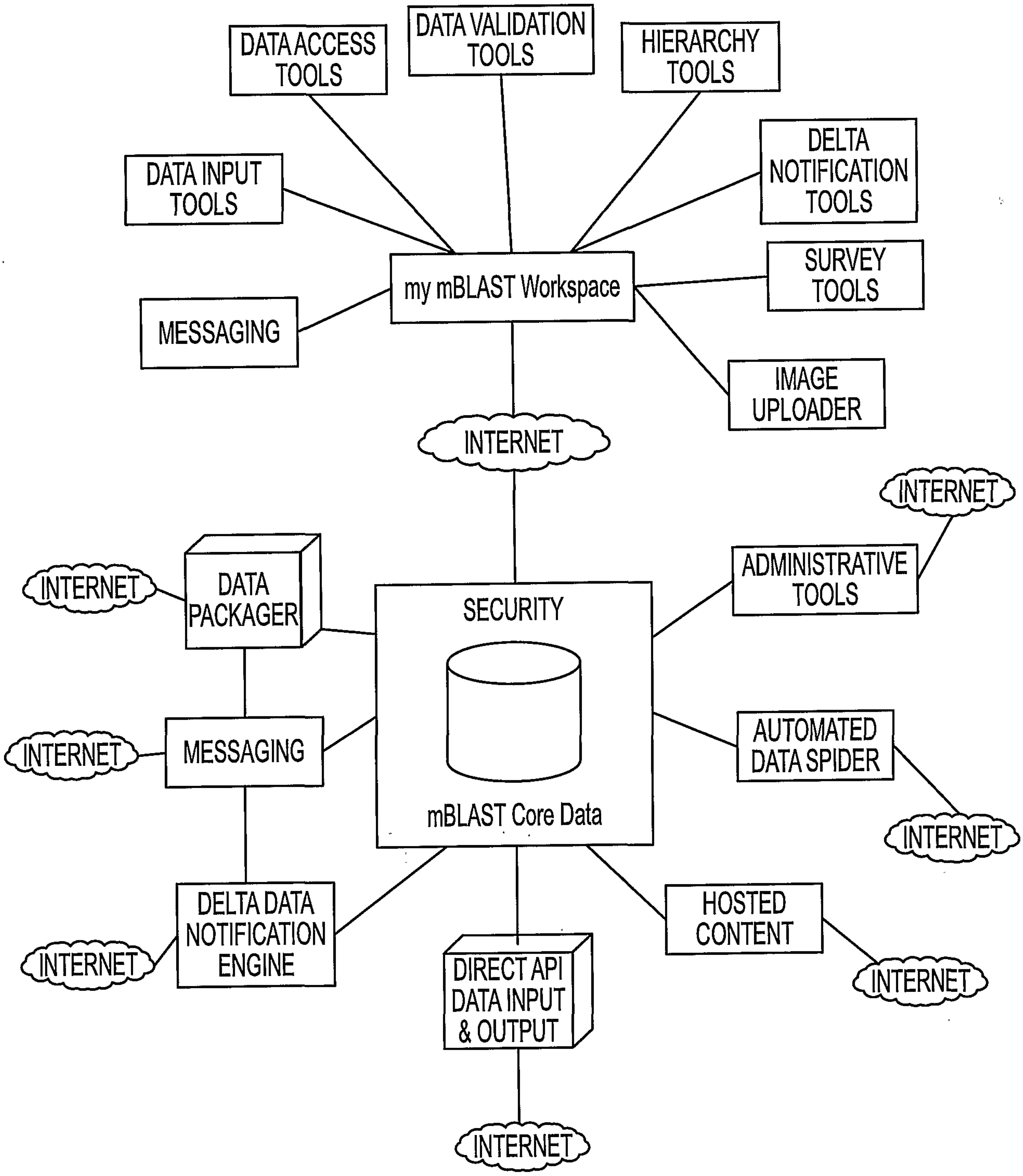
DATA ACCESS TOOLS
DATA VALIDATION TOOLS
HIERARCHY TOOLS
DATA INPUT TOOLS
DELTA NOTIFICATION TOOLS
my mBLAST Workspace
SURVEY TOOLS
MESSAGING
IMAGE UPLOADER
INTERNET
INTERNET
INTERNET
DATA PACKAGER
ADMINISTRATIVE TOOLS
SECURITY
INTERNET
MESSAGING
AUTOMATED DATA SPIDER
mBLAST Core Data
INTERNET
DELTA DATA NOTIFICATION ENGINE
HOSTED CONTENT
INTERNET
DIRECT API DATA INPUT & OUTPUT
INTERNET
INTERNET

FIG. 1

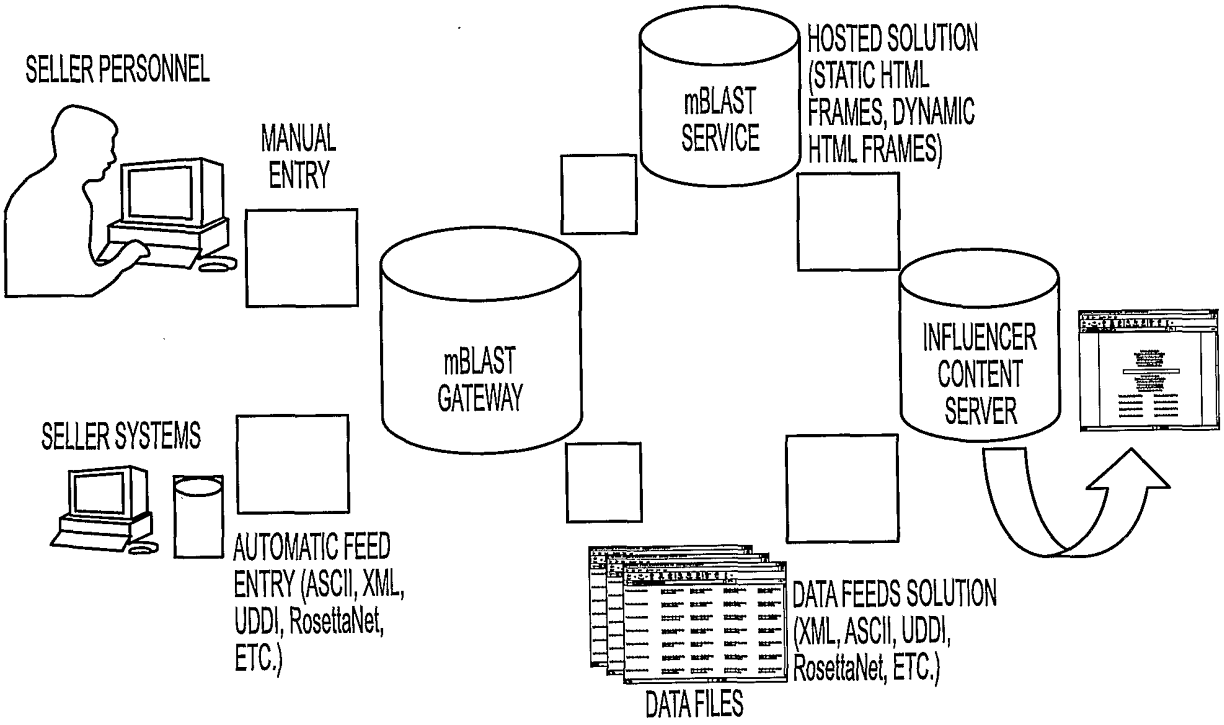
SELLER PERSONNEL
MANUAL ENTRY
mBLAST SERVICE
HOSTED SOLUTION (STATIC HTML FRAMES, DYNAMIC HTML FRAMES)
mBLAST GATEWAY
INFLUENCER CONTENT SERVER
SELLER SYSTEMS
AUTOMATIC FEED ENTRY (ASCII, XML, UDDI, RosettaNet, ETC.)
DATA FEEDS SOLUTION (XML, ASCII, UDDI, RosettaNet, ETC.)
DATA FILES

FIG. 2

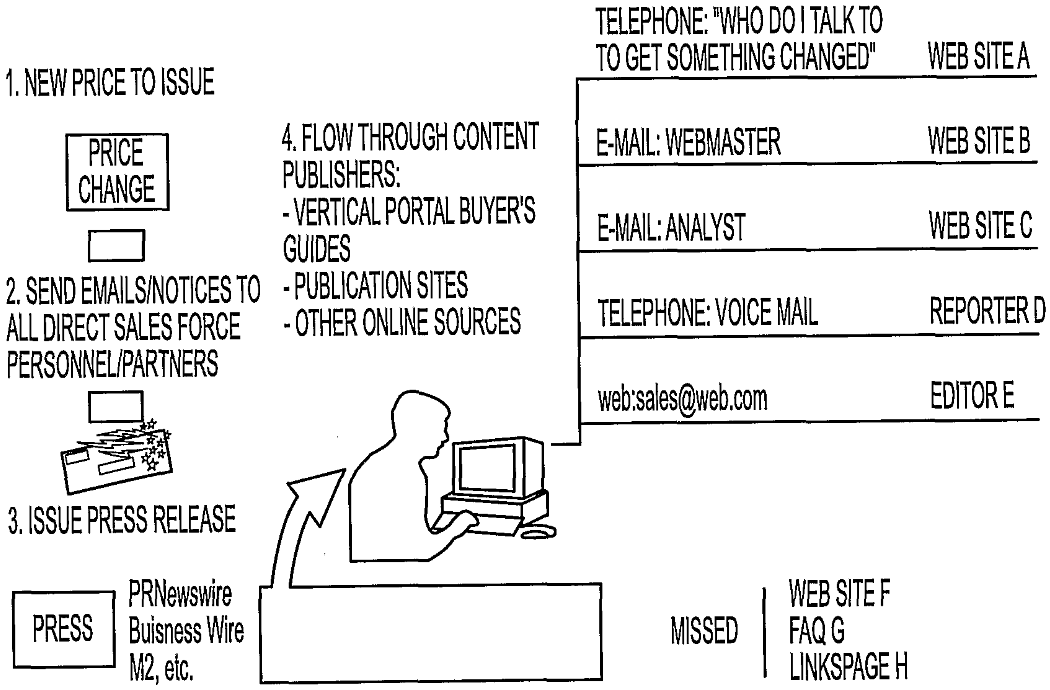
1. NEW PRICE TO ISSUE
PRICE CHANGE
2. SEND EMAILS/NOTICES TO ALL DIRECT SALES FORCE PERSONNEL/PARTNERS
3. ISSUE PRESS RELEASE
PRESS
PRNewswire
Buisness Wire
M2, etc.
4. FLOW THROUGH CONTENT PUBLISHERS:
- VERTICAL PORTAL BUYER'S GUIDES
- PUBLICATION SITES
- OTHER ONLINE SOURCES
TELEPHONE: "WHO DO I TALK TO TO GET SOMETHING CHANGED"
WEB SITE A
E-MAIL: WEBMASTER
WEB SITE B
E-MAIL: ANALYST
WEB SITE C
TELEPHONE: VOICE MAIL
REPORTER D
web:sales@web.com
EDITOR E
MISSED
WEB SITE F
FAQ G
LINKSPAGE H

FIG. 3

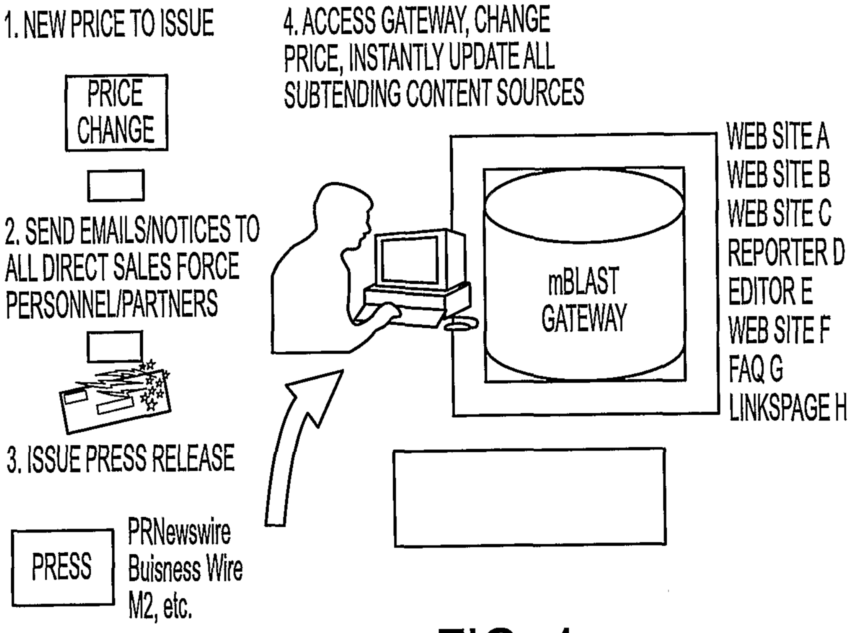
1. NEW PRICE TO ISSUE
PRICE CHANGE
2. SEND EMAILS/NOTICES TO ALL DIRECT SALES FORCE PERSONNEL/PARTNERS
3. ISSUE PRESS RELEASE
PRESS
PRNewswire
Buisness Wire
M2, etc.
4. ACCESS GATEWAY, CHANGE PRICE, INSTANTLY UPDATE ALL SUBTENDING CONTENT SOURCES
mBLAST GATEWAY
WEB SITE A
WEB SITE B
WEB SITE C
REPORTER D
EDITOR E
WEB SITE F
FAQ G
LINKSPAGE H

FIG. 4

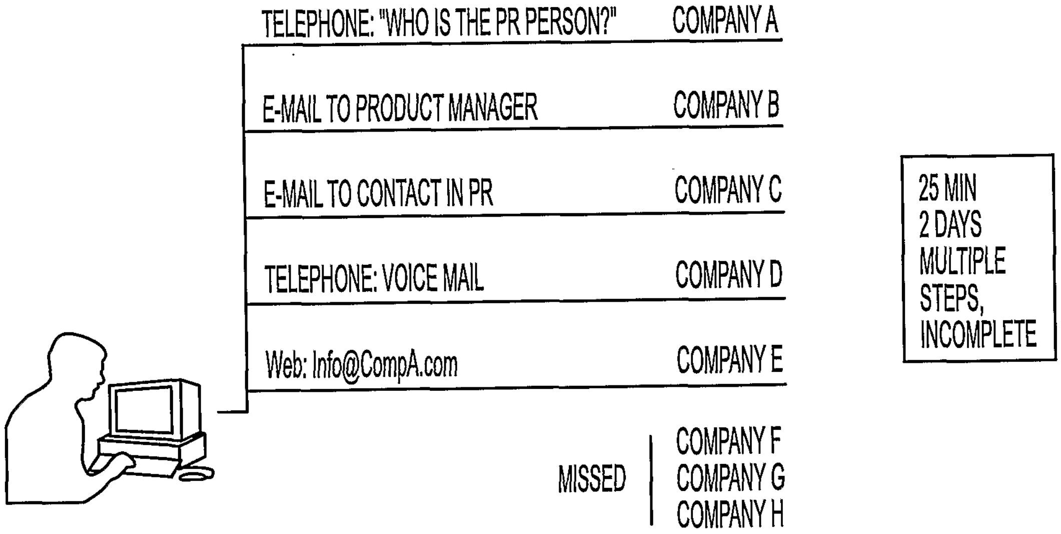
TELEPHONE: "WHO IS THE PR PERSON?"
COMPANY A
E-MAIL TO PRODUCT MANAGER
COMPANY B
E-MAIL TO CONTACT IN PR
COMPANY C
TELEPHONE: VOICE MAIL
COMPANY D
Web: Info@CompA.com
COMPANY E
MISSED
COMPANY F
COMPANY G
COMPANY H
25 MIN
2 DAYS
MULTIPLE
STEPS,
INCOMPLETE

FIG. 5

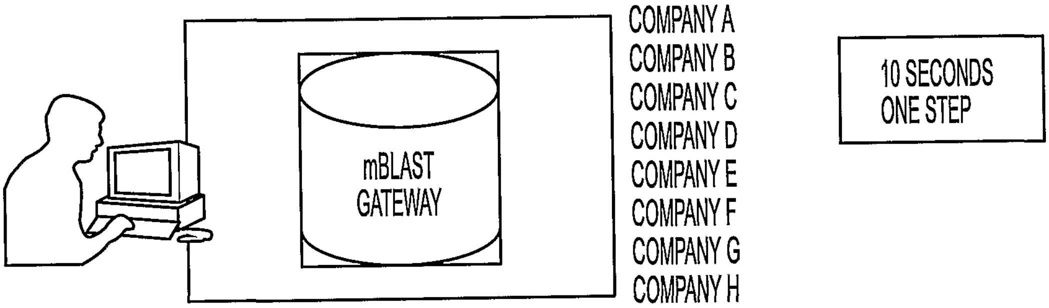
mBLAST
GATEWAY
COMPANY A
COMPANY B
COMPANY C
COMPANY D
COMPANY E
COMPANY F
COMPANY G
COMPANY H
10 SECONDS
ONE STEP

FIG. 6

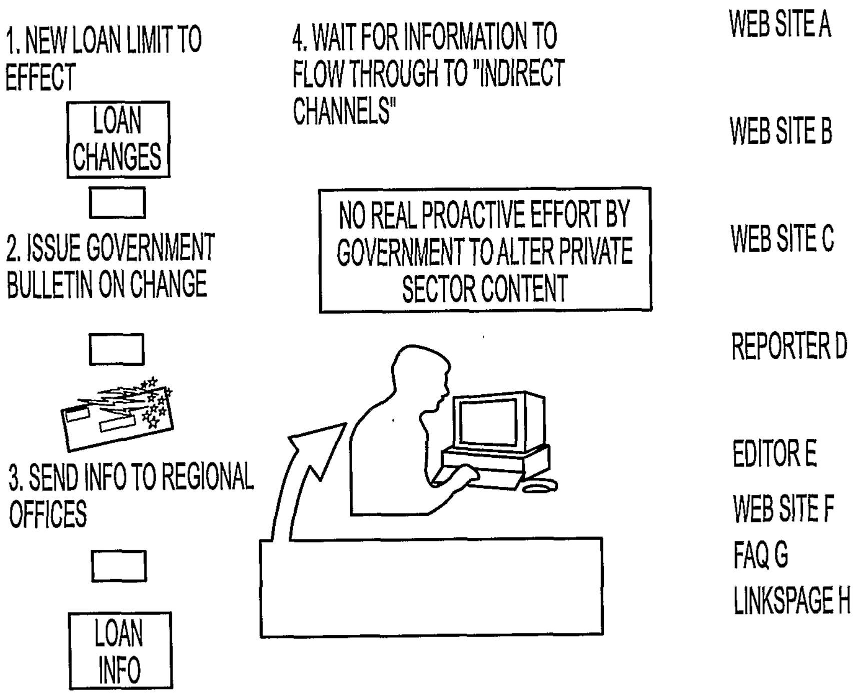
1. NEW LOAN LIMIT TO EFFECT
LOAN CHANGES
2. ISSUE GOVERNMENT BULLETIN ON CHANGE
3. SEND INFO TO REGIONAL OFFICES
LOAN INFO
4. WAIT FOR INFORMATION TO FLOW THROUGH TO "INDIRECT CHANNELS"
NO REAL PROACTIVE EFFORT BY GOVERNMENT TO ALTER PRIVATE SECTOR CONTENT
WEB SITE A
WEB SITE B
WEB SITE C
REPORTER D
EDITOR E
WEB SITE F
FAQ G
LINKSPAGE H

FIG. 7

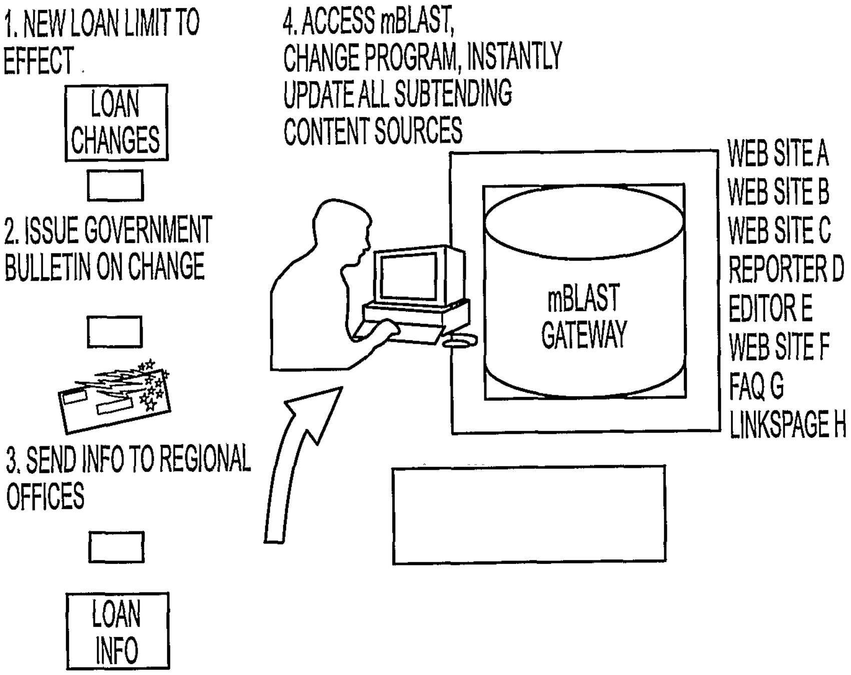
1. NEW LOAN LIMIT TO EFFECT
LOAN CHANGES
2. ISSUE GOVERNMENT BULLETIN ON CHANGE
3. SEND INFO TO REGIONAL OFFICES
LOAN INFO
4. ACCESS mBLAST, CHANGE PROGRAM, INSTANTLY UPDATE ALL SUBTENDING CONTENT SOURCES
mBLAST GATEWAY
WEB SITE A
WEB SITE B
WEB SITE C
REPORTER D
EDITOR E
WEB SITE F
FAQ G
LINKSPAGE H

FIG. 8

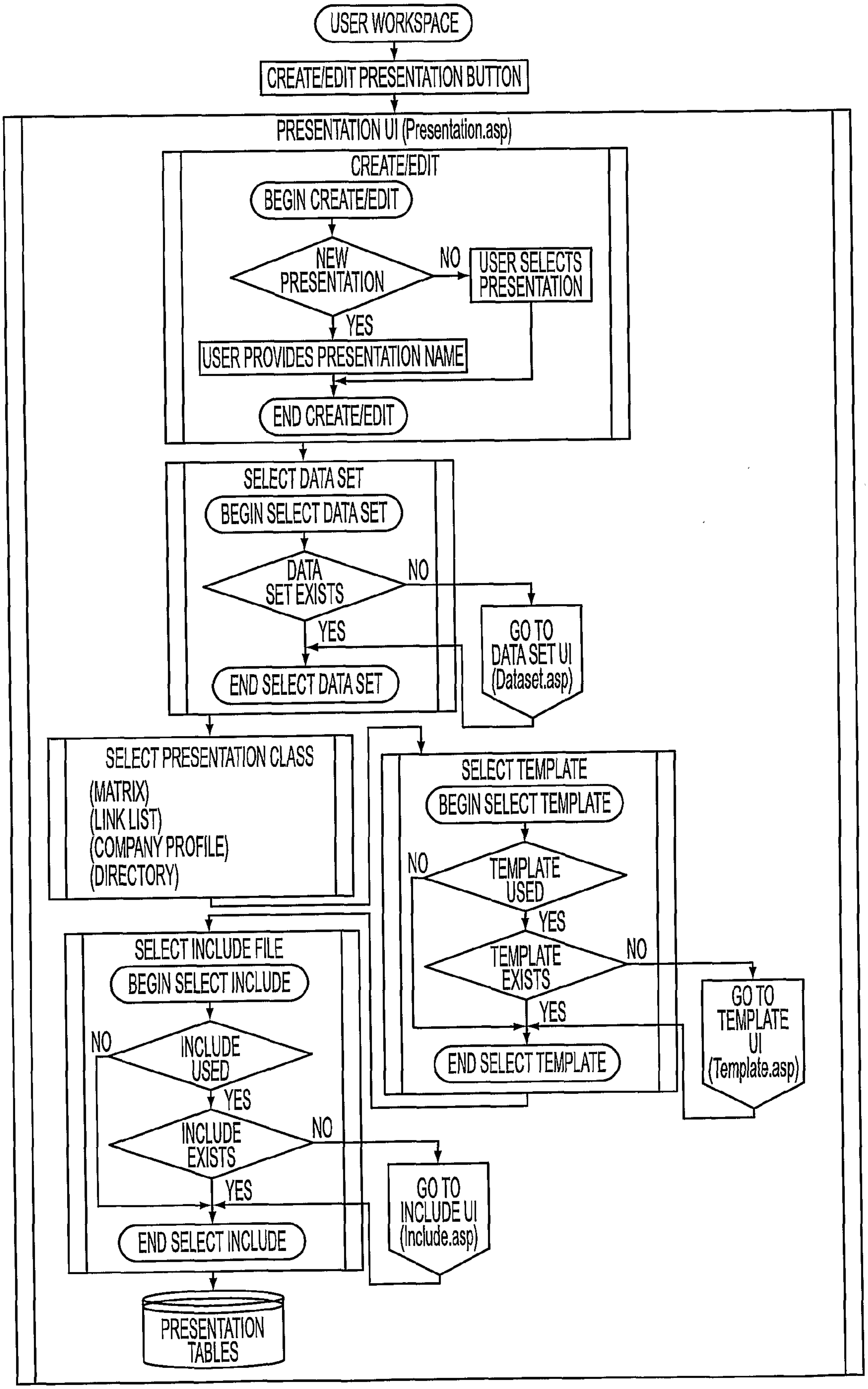
USER WORKSPACE
CREATE/EDIT PRESENTATION BUTTON
PRESENTATION UI (Presentation.asp)
CREATE/EDIT
BEGIN CREATE/EDIT
NEW PRESENTATION
NO
USER SELECTS PRESENTATION
YES
USER PROVIDES PRESENTATION NAME
END CREATE/EDIT
SELECT DATA SET
BEGIN SELECT DATA SET
DATA SET EXISTS
NO
GO TO DATA SET UI (Dataset.asp)
YES
END SELECT DATA SET
SELECT PRESENTATION CLASS
(MATRIX)
(LINK LIST)
(COMPANY PROFILE)
(DIRECTORY)
SELECT TEMPLATE
BEGIN SELECT TEMPLATE
NO
TEMPLATE USED
YES
TEMPLATE EXISTS
NO
GO TO TEMPLATE UI (Template.asp)
YES
END SELECT TEMPLATE
SELECT INCLUDE FILE
BEGIN SELECT INCLUDE
NO
INCLUDE USED
YES
INCLUDE EXISTS
NO
GO TO INCLUDE UI (Include.asp)
YES
END SELECT INCLUDE
PRESENTATION TABLES

FIG. 9

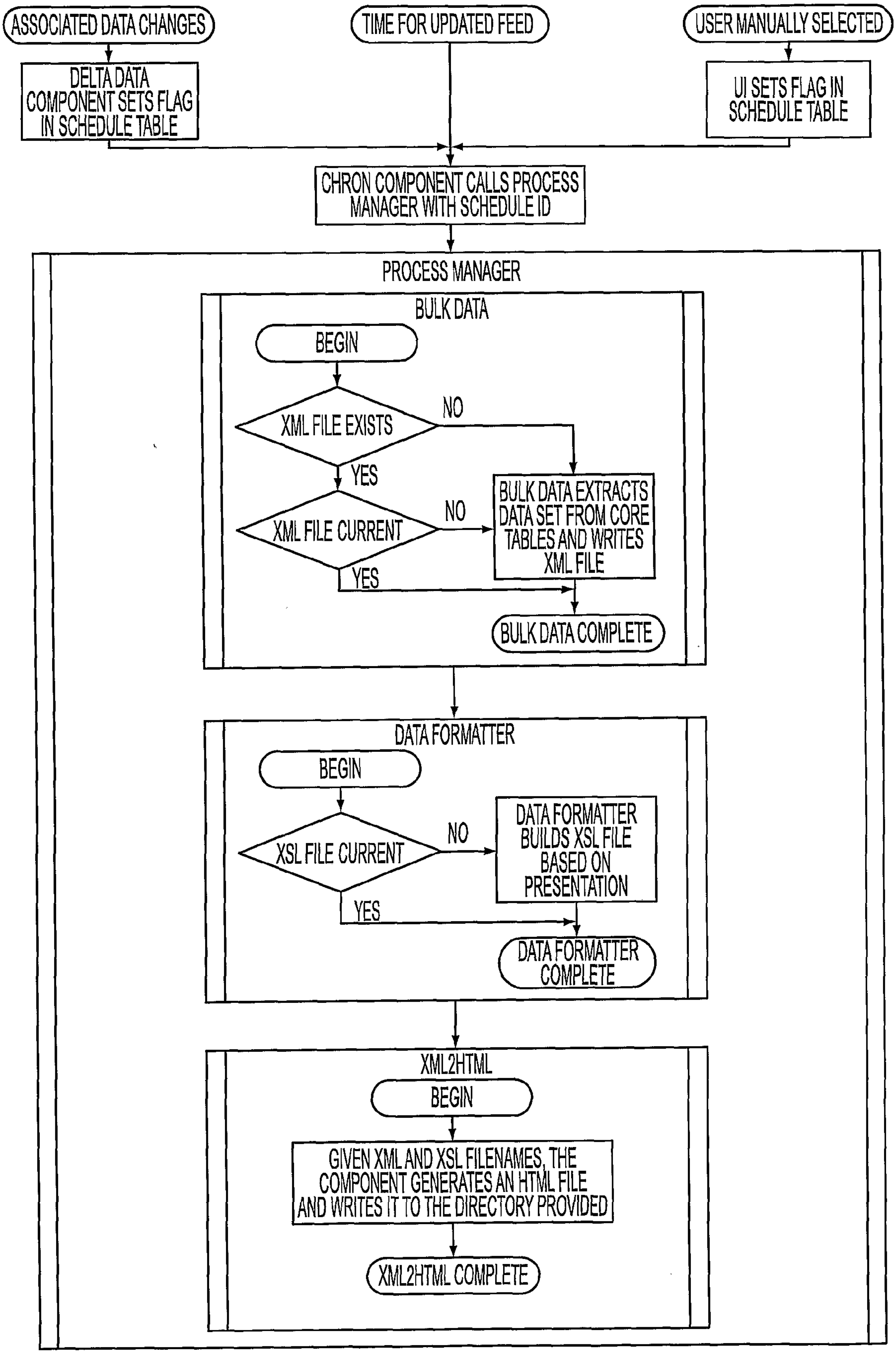
ASSOCIATED DATA CHANGES
DELTA DATA COMPONENT SETS FLAG IN SCHEDULE TABLE
TIME FOR UPDATED FEED
USER MANUALLY SELECTED
UI SETS FLAG IN SCHEDULE TABLE
CHRON COMPONENT CALLS PROCESS MANAGER WITH SCHEDULE ID
PROCESS MANAGER
BULK DATA
BEGIN
XML FILE EXISTS
NO
YES
XML FILE CURRENT
NO
YES
BULK DATA EXTRACTS DATA SET FROM CORE TABLES AND WRITES XML FILE
BULK DATA COMPLETE
DATA FORMATTER
BEGIN
XSL FILE CURRENT
NO
YES
DATA FORMATTER BUILDS XSL FILE BASED ON PRESENTATION
DATA FORMATTER COMPLETE
XML2HTML
BEGIN
GIVEN XML AND XSL FILENAMES, THE COMPONENT GENERATES AN HTML FILE AND WRITES IT TO THE DIRECTORY PROVIDED
XML2HTML COMPLETE

FIG. 10

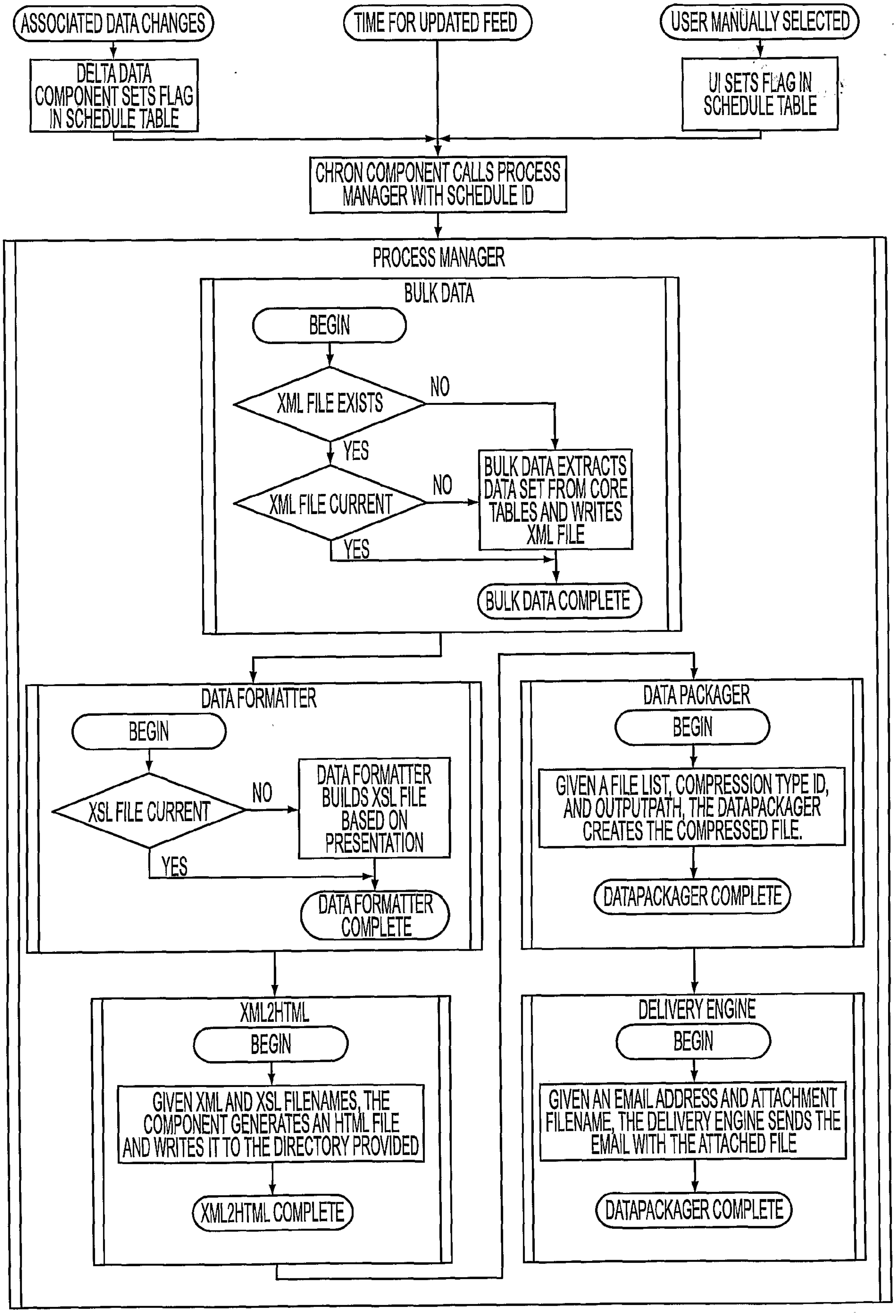
ASSOCIATED DATA CHANGES
TIME FOR UPDATED FEED
USER MANUALLY SELECTED
DELTA DATA COMPONENT SETS FLAG IN SCHEDULE TABLE
UI SETS FLAG IN SCHEDULE TABLE
CHRON COMPONENT CALLS PROCESS MANAGER WITH SCHEDULE ID
PROCESS MANAGER
BULK DATA
BEGIN
XML FILE EXISTS
NO
YES
XML FILE CURRENT
NO
YES
BULK DATA EXTRACTS DATA SET FROM CORE TABLES AND WRITES XML FILE
BULK DATA COMPLETE
DATA FORMATTER
BEGIN
XSL FILE CURRENT
NO
YES
DATA FORMATTER BUILDS XSL FILE BASED ON PRESENTATION
DATA FORMATTER COMPLETE
DATA PACKAGER
BEGIN
GIVEN A FILE LIST, COMPRESSION TYPE ID, AND OUTPUTPATH, THE DATAPACKAGER CREATES THE COMPRESSED FILE.
DATAPACKAGER COMPLETE
XML2HTML
BEGIN
GIVEN XML AND XSL FILENAMES, THE COMPONENT GENERATES AN HTML FILE AND WRITES IT TO THE DIRECTORY PROVIDED
XML2HTML COMPLETE
DELIVERY ENGINE
BEGIN
GIVEN AN EMAIL ADDRESS AND ATTACHMENT FILENAME, THE DELIVERY ENGINE SENDS THE EMAIL WITH THE ATTACHED FILE
DATAPACKAGER COMPLETE

FIG. 11