A system (60) for manipulating information regarding a product in a client/server computer system is provided. The system includes a client computer (65) operable to run a PSCM client component (68) and is coupled to a server computer (62). The server side (62) comprises a repository (80) operable to store a plurality of view objects representing at least one product, a handles database (78) operable to store a plurality of handles, wherein each handle (79) is associated with a view object stored in a repository; and a PSCM server (74) operable to interpret requests from the client computer (65) and return a derived configuration object to the client (65) using the handle (79).
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>AL</td>
<td>Albania</td>
<td>ES</td>
<td>Spain</td>
<td>LS</td>
<td>Lesotho</td>
<td>SI</td>
<td>Slovenia</td>
</tr>
<tr>
<td>AM</td>
<td>Armenia</td>
<td>FI</td>
<td>Finland</td>
<td>LT</td>
<td>Lithuania</td>
<td>SK</td>
<td>Slovakia</td>
</tr>
<tr>
<td>AT</td>
<td>Austria</td>
<td>FR</td>
<td>France</td>
<td>LU</td>
<td>Luxembourg</td>
<td>SN</td>
<td>Senegal</td>
</tr>
<tr>
<td>AU</td>
<td>Australia</td>
<td>GA</td>
<td>Gabon</td>
<td>LV</td>
<td>Latvia</td>
<td>SZ</td>
<td>Swaziland</td>
</tr>
<tr>
<td>AZ</td>
<td>Azerbaijan</td>
<td>GB</td>
<td>United Kingdom</td>
<td>MC</td>
<td>Monaco</td>
<td>TD</td>
<td>Chad</td>
</tr>
<tr>
<td>BA</td>
<td>Bosnia and Herzegovina</td>
<td>GE</td>
<td>Georgia</td>
<td>MD</td>
<td>Republic of Moldova</td>
<td>TG</td>
<td>Togo</td>
</tr>
<tr>
<td>BB</td>
<td>Barbados</td>
<td>GH</td>
<td>Ghana</td>
<td>MG</td>
<td>Madagascar</td>
<td>TJ</td>
<td>Tajikistan</td>
</tr>
<tr>
<td>BF</td>
<td>Burkina Faso</td>
<td>GN</td>
<td>Guinea</td>
<td>MK</td>
<td>The former Yugoslav Republic of Macedonia</td>
<td>TR</td>
<td>Turkmenistan</td>
</tr>
<tr>
<td>BG</td>
<td>Bulgaria</td>
<td>GR</td>
<td>Greece</td>
<td>ML</td>
<td>Mali</td>
<td>TT</td>
<td>Trinidad and Tobago</td>
</tr>
<tr>
<td>BJ</td>
<td>Benin</td>
<td>HU</td>
<td>Hungary</td>
<td>MN</td>
<td>Mongolia</td>
<td>UA</td>
<td>Ukraine</td>
</tr>
<tr>
<td>BR</td>
<td>Brazil</td>
<td>IE</td>
<td>Ireland</td>
<td>MR</td>
<td>Mauritania</td>
<td>UG</td>
<td>Uganda</td>
</tr>
<tr>
<td>BY</td>
<td>Belarus</td>
<td>IL</td>
<td>Israel</td>
<td>MW</td>
<td>Malawi</td>
<td>US</td>
<td>United States of America</td>
</tr>
<tr>
<td>CA</td>
<td>Canada</td>
<td>IS</td>
<td>Iceland</td>
<td>MX</td>
<td>Mexico</td>
<td>UZ</td>
<td>Uzbekistan</td>
</tr>
<tr>
<td>CF</td>
<td>Central African Republic</td>
<td>JP</td>
<td>Japan</td>
<td>NE</td>
<td>Niger</td>
<td>VN</td>
<td>Viet Nam</td>
</tr>
<tr>
<td>CG</td>
<td>Congo</td>
<td>KE</td>
<td>Kenya</td>
<td>NL</td>
<td>Netherlands</td>
<td>YU</td>
<td>Yugoslavia</td>
</tr>
<tr>
<td>CH</td>
<td>Switzerland</td>
<td>KG</td>
<td>Kyrgyzstan</td>
<td>NO</td>
<td>Norway</td>
<td>ZW</td>
<td>Zimbabwe</td>
</tr>
<tr>
<td>CI</td>
<td>Côte d’Ivoire</td>
<td>KP</td>
<td>Democratic People’s Republic of Korea</td>
<td>NZ</td>
<td>New Zealand</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CM</td>
<td>Cameroon</td>
<td>KR</td>
<td>Republic of Korea</td>
<td>PL</td>
<td>Poland</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CN</td>
<td>China</td>
<td>KZ</td>
<td>Kazakhstan</td>
<td>PT</td>
<td>Portugal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CU</td>
<td>Cuba</td>
<td>LC</td>
<td>Saint Lucia</td>
<td>RO</td>
<td>Romania</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CZ</td>
<td>Czech Republic</td>
<td>LI</td>
<td>Liechtenstein</td>
<td>RU</td>
<td>Russian Federation</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DE</td>
<td>Germany</td>
<td>LK</td>
<td>Sri Lanka</td>
<td>SD</td>
<td>Sudan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>DK</td>
<td>Denmark</td>
<td>LR</td>
<td>Liberia</td>
<td>SE</td>
<td>Sweden</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EE</td>
<td>Estonia</td>
<td></td>
<td></td>
<td>SG</td>
<td>Singapore</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**FOR THE PURPOSES OF INFORMATION ONLY**

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.
PRODUCT STRUCTURE CONFIGURATION MANAGEMENT
SERVICE SYSTEM ON COMPUTER NETWORKS

TECHNICAL FIELD OF THE INVENTION

This invention relates to computer aided product design, and more particularly to a product structure configuration management service system on computer networks.

BACKGROUND OF THE INVENTION

Product data management (PDM) refers to methods for managing product design and manufacture. An example of a PDM system is the Information Manager (IMAN) sold by Unigraphics Solutions Incorporated. IMAN is designed as an enterprise-wide product data management (PDM) system encompassing all phases of product life cycles. One focus of the system is to represent the enterprise in terms of objects and operations on objects. Different object classes can be formed by modeling enterprise activities such as design, manufacture, and administration. Other popular computer aided systems include computer aided design (CAD) and computer aided manufacturing systems (CAM). These are used to design and revise design of products.

PDM and CAD/CAM systems are becoming increasingly important in the manufacturing sector. Several improvements in PDM systems have been achieved. One such improvement is disclosed in Patent No. 5,434,791, entitled Product Structure Management (PSM), by Boma Koko and assigned to Electronic Data Systems, Inc. This patent discloses a PDM system which allows different users the ability to view different configurations of a product line. However, these PDM systems are typically proprietary, requiring special software on both the
client and server side. Additionally, organizations with computer systems such as legacy systems, for example, need to access PDM systems. However, the conversion to a proprietary system can be very expensive when older databases need to be converted to work with newer PDM systems. What is needed is a product structure configuration management system for users of heterogeneous hardware and software platforms.

SUMMARY OF THE INVENTION

Accordingly, it may be appreciated that a need has arisen for a product structure configuration management (PSCM) service system on computer networks. In accordance with the teachings of the present invention, a method and system are provided which substantially eliminates or reduces disadvantages and problems associated with existing product data management systems.

In one embodiment, a method for managing information regarding a product in a client/server computer system is provided. In one step of the method a model of at least one product is stored as a plurality of view objects in a repository at a server. In the next step, each view object is associated with a handle where each handle contains information about each view object. In the third step, each handle is stored in a handle database along with partial configuration rules at a server. In the fourth step, a request for information is received from a client computer. Then, in the fifth step, a configuration is formed based on the request. In the sixth step a derived configuration object is returned to the client utilizing the handles and partial rules stored in the handles database of the repository along with information in the request.
In another embodiment, a system for manipulating information regarding a product in a client/server computer system is disclosed. The system includes a client computer operable to run a product structure configuration management (PSCM) front end program coupled to a server computer. The server computer comprises a repository operable to store a plurality of view objects representing at least one product, a handle database operable to store a plurality of handles, wherein each handle is associated with a view object stored in a repository; and a PSCM server operable to interpret requests from the client computer and return a derived configuration object to the client using the handles.

The present invention provides various technical advantages over current product data management systems. For example, one technical advantage is that a product structure management system can be accessed and utilized by a variety of clients using a variety of non-proprietary software. Another technical advantage is that the common server repository may be accessed by other clients. Other technical advantages may be readily apparent to one skilled in the art from the following figures, descriptions, and claims.

**BRIEF DESCRIPTION OF THE DRAWINGS**

For a more complete understanding of the present invention and the advantages thereof, reference is now made to the following descriptions taken into conjunction with the accompanying drawings, in which like reference numbers represent like parts, in which:

FIGURE 1 illustrates an object model of the product structure configuration management system; and
FIGURE 2 illustrates the architecture of the product structure configuration management system in accordance with the teaching of the present invention;

FIGURE 3 is a flowchart outlining the operation of the PSCM system in accordance with the teaching of the present invention; and,

FIGURE 4 illustrates the application program interface for the PSCM system in accordance with the teachings of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

In order to understand the following figures and accompanying discussion, it is important to understand the definition of some of the terms used in product data management. Therefore, the following is a brief description of important terms and their definitions as used in the present invention.

An item represents an entity, whether physical or conceptual, that a business enterprise is interested in managing a life cycle of revisions and usages of based on various contexts. An item can range from a document that an enterprise wishes to maintain a history of to a physical item that is to be integrated into a product. An item revision is a particular definition of the item at some point in time during its life cycle. For example, if the item is a document, an item revision might be the document as it existed at a certain point in time, i.e., after a review or after several reviews. An item typically has multiple item revisions, each representing how the item existed at a certain point in time. The focus of the product structure configuration manager (PSCM) is not in what is included in the item revisions, but rather how to manage the relationship between items and item revisions such that any user using
any software can track the changes in an item. An item view represents aspects of an item developed by different disciplines. That is, if an item is a grouping of a blueprint of a product along with a description, an item view might be the blueprint as it was developed by drafting, whereas a description of how that blueprint is assembled might be a different item view. Since item views may change over time, item views also have associated with them item view revisions. Together, these are often referred to as view objects. Finally, a product is the tangible results of a collaborative effort or the parts of such a result. This can range anywhere from a complete product such as a car or the final assembly of a document including art work, spread sheets or any other necessary components. Products will be defined by what an enterprise or application actually needs to make a product.

A configuration is an assembly of configuration elements that satisfies a set of configuration rules for a complete assembly. A configuration element may reference items, item revisions, item views or item view revisions that satisfies the rules for membership in the configuration. A configuration rule is a rule that governs membership in the configuration. All configuration elements in the configuration must satisfy the rules. In operation, if a user wants to retrieve an item, item revision, item view, or item view revision stored in a database, the user may query the database and ask for a specific product to be displayed. In the case of a computer manufacturer, the user may request that all information involving a computer product known as the "PC2000" be displayed. That request would then trigger a certain number of configuration rules. These rules would detail what parts go into the PC2000. The
configuration rules would then be used by a program (the product structure configuration manager, discussed in conjunction with Figure 1) that will locate items, item revisions, item views, item view revisions that satisfy those rules. The assembly of those configuration elements would make up the configuration. Therefore, as an example, after the query by the user, a configuration may be returned that includes a certain type of monitor, keyboard, computer with a certain processor and memory requirement. Additionally, there can be configuration objects and configuration object revisions. A configuration object relates a set of configurations in a sequence of evolutionary changes. A configuration object revision stores a time-dependent configuration such that comparisons can be made between revisions. In the case of the preceding example, over time, the components involved in the PC2000 may change. As such, there may be multiple configuration object revisions.

FIGURE 1 illustrates an object model for the product structure configuration management (PSCM) system. First, a configuration object 10 is shown. Configuration object 10 is a complete set of configuration object revisions. Associated with each configuration object 10 is a revision history 12 and zero or more derived configuration objects 16. Revision history 12 contains information on the revisions to parts of the configuration object 10. Each revision history 12 has at least one revision element 14, referencing one configuration object revision 18 of configuration object 10. Revision element 14 includes administrative information about a configuration object revision 18. A configuration object revision 18 is a configuration object 10 at some point in time, similar to an item revision. Associated with configuration object revision

SUBSTITUTE SHEET (RULE 26)
18 are configuration rules 20. Configuration rules 20 are used to assemble the configuration and restrict its use in other configuration. Every configuration object revision 18 contains one or more configuration elements 22 which makes up configuration object revision 18.

Configuration element 22 may reference the following view objects: an item 24, an item view 26, an item view revision 28, an external object revision 44 or even a configuration object 10. Item 24 is an imprecise object that can change with time. Item 24 can be one or more documents, drawings, models or parts of products that an enterprise wishes to maintain over some life cycle. Associated with item 24 are alternate items 32. If, for example, item 24 is a type of memory chip, alternate items 32 may be a memory chip from a different manufacturer. Item 24 is also associated with a revision history 30, that tracks changes of items over time. Item view 26 is a view specific to particular users, such as the aspects of an item 24 that would involve a draftsman versus the aspects of items that concern a plant manager. Item views 26 also changes with time and are associated with a revision history 30. Item view revision 28 is the revision of item view 26 and is precise in the sense it is a specific aspect of an item at a specific time. An external object revision 44 is information provided for which the revision history is external, or maintained by some other system to PSCM system such as proprietary algorithms for responding to trade secret requirements.

Associated with each item view revision 28 is an optional product structure 34 which represents an assembled product. Associated with product structure 34 is a component 36 that can be either precise 31 (a fully evaluated product), imprecise 42 (a not fully evaluated product) or a version of a component 40.
A typed attachment 50 may be a specification or a requirement with associated business rules which can be embedded in the server. These include information such as governmental warnings or directives, design, requirements or functional specifications. Attachment 50 may also relate a part of a product with the product by attaching the part to the product or to another part. Attachment 50 may reference items 52, item revisions 54, item views 48, item view revisions 56, and external object revisions.

FIGURE 2 illustrates the architecture of the product structure configuration management (PSCM) system 60 in accordance with the teaching of the present invention. Illustrated is PSCM system 60 comprising a client side 61 and a server side 62. Client side 61 comprises a client computer 65 which is operable to run a variety of PSCM client applications 66, such as CAD/CAM programs, web browsers, PDM programs and document management programs. Computer 65 can be any conventional computer system running on any of a number of operating systems, such as UNIX or WINDOWS/WINDOWS NT. Computer 65 can also run other programs not illustrated such as word processing and database management programs. Computer 65 includes a variety of inputs 63 and outputs 64. Input 63 can be inputs from external file servers, keyboards, scanners or other input devices. Output 64 can be any output including an output to a printer, another network, a connection to another computer or network via a modem or any other output. Computer 65 also includes a PSCM client component 68. Interface 70 represents a division between client side 61 and server side 62. Interface 70 may represent a local area network connection, a wide area network connection, a connection through the Internet, a connection through a corporate intranet, or
any other way of connecting a client and server system. Communication across interface 70 may be assisted by facilities such as the Java programming language, object linking embedding (OLE), activeX components, or CORBA and appropriate hardware such as routers, switches, and cables.

PSCM client applications 66 are in communication with several server side applications. PSCM client component 68 is bound to PSCM client applications 66 which communicate with other systems such as a distributed file server (DFS) 72 and a standard query language (SQL) server 76. DFS 72 is a generic file server and is in communication with an intellectual property repository 80. SQL server 76 is a generic database server and is also in communication with an intellectual property repository 80. PSCM client component 68 is in communication with a PSCM server 74. PSCM server 74 is a server specific to the present invention as described in this application. It is accessible by client computer 65 regardless of the software client computer 65 is running. PSCM server 74 is coupled to a Configuration Manager/Data Store (CM/DS) 78 using standard query language (SQL), or other appropriate means. CM/DS 78 is a part of intellectual property repository 80. CM/DS 78 contains, among other things, a plurality of handles 79. Handles 79 contain information regarding the object views stored in intellectual property repository 80. Handles 79 store information which will identify the associated object in the repository 80 and are maintained in CM/DS 78. Handles 79 also store information regarding revisions and can optionally contain other descriptive fields as well.

PSCM server 74 is accessible by any user able to interface with server side 62. No special software is
needed to access PSCM server 74. CM/DS 78 also contains a variety of configuration rules 20 which are stored for easy access by PSCM server 74. Intellectual property repository 80 stores all of the view objects such as items 24, item views 26, and item view revisions 28, as well as storing configurations objects 10 as needed. The efficiency of PSCM system 10 is increased by maintaining a separate database for handles 79. In this way, operations on handles 79 as well as other parameters stored in CM/DS 78 can be done by PSCM server 74 and the results assembled without having to actually manipulate the entire files as stored in the repository. DFS server 72 and SQL server 76, as mentioned before, are in communication with intellectual property repository 80.

PSCM server 74 is also coupled to an update manager 75. Update manager 75 monitors changes to configuration objects 10 that are associated with derived configuration objects 16. In one embodiment update manager 75 operates by checking each object stored with PSCM server 74 when view objects are saved or changed and checks to see if any derived configuration object 16 is based on this changed object. If so, update manager flags derived configuration object 16 as out of date. Alternatively, update manager 75 could reevaluate derived configuration object 16. In either case, update manager 75 is operable to send electronic mail to any party affected by the change. Update manager 75 may be programmed to take other action, based on events occurring in PSCM server 74. An advantage of update manager 75 is that it allows updates to derived configuration objects 16 to occur automatically. This increases the efficiency of the systems by storing updated and fully evaluated derived configuration object 16 prepared before a user requests one.
In operation, a user at client computer 65 may be operating one or more programs related to the user's job in developing a product. For example, the user may be a design engineer running a CAD program to update a set of drawings for a computer case that will be used in the "PC 2000" product. At the same point in time the user is doing this activity, another user, a technical writer, could be at another client computer writing the documentation for the "PC 2000" product. Already stored in repository 80 are all items 24 associated with the "PC 2000" product, including the original versions of the computer case drawing and the documentation. When the designer is finished with the drawings, it is saved as an item view revision 28 in repository 80. A handle in CM/DS 78 will be created to denote the new revision. When the technical writer is finished with the changes to the document, it is saved as an item view revision 28 in repository 80 and registered with PSCM server 74 which then creates another new handle 79 to represent the new revision in the handles database. Since the drawings and the documentation are associated at some level, that association is stored as an attachment 50 in CM/DS 78.

A third user at another client computer 65 may be a project manager. Using a web browser or some other type of software, the project manager may make a request to see all specifications for the "PC 2000". This request would go through PSCM client 68, to PSCM server 74 where it would be broken down into configuration rules 20. PSCM server 74 would then evaluate these rules to find specific items that satisfy these rules 20. The rules would be executed on handles 79 stored in CM/DS 78 and the results would point to specific object views such as item view revisions 28 to be retrieved from repository 80. In this case, the CAD drawings of the computer case
would be retrieved. Since this is an item view revision 28, the revision would be retrieved. The documentation would also be retrieved. Other items would also be retrieved. All the items that satisfy configuration rules 20 are known as evaluated configuration elements 21. All the elements together would then give a derived configuration object 16. Derived configuration object 16 would be returned to the requesting party.

A user at client computer can also request information from repository 80 without triggering PSCM server 74. For example, a user may retrieve a specific file through DFS 72, or a database lookup could be done using SQL server 78. Thus, the system provides users the ability to access PSCM server 74 using a variety of third-party software as well as the ability to access repository 80 for a variety of different applications.

FIGURE 3 is a flowchart outlining the operation of the PSCM system 10 in accordance with the teaching of the present invention. In step 90, items 24, item views 26, item view revision 28, configuration rules 20. and configuration objects 10 are stored in repository 80 as they are produced by PSCM server during operation. In step 92, CM/DS 78 is configured with the associated handles 79. In step 94, handles 79 are associated with the view objects stored in the repository, along with any configuration object revisions 18 which are related to configuration objects 10. This step also associates configuration rules 20 with configuration object revisions 18 as necessary. Of course, being a dynamic system, as other item view revisions 28, configuration revision 18, and other updates are developed, there will also be stored in the repository 80. Associated handles 79 are stored in CM/DS 78.
In step 96, a client request is received by PSCM server 74. As discussed above, the client request might be a request to display all documents for a particular product. The request might be to view all drawings for a document - an item view 26 request. Any number of requests might be received from a user by PSCM server 74. PSCM server 74 then interprets the request and initiates the evaluation of one or more stored configuration object 10 using handles 79. In step 98, the requests are analyzed and information contained in them is used to evaluate configuration rules 20 and determine the configuration using handles 79. Finally, in step 99, the appropriate derived configuration object 16 which contains all the qualifying elements from step 98 is returned to the user. A derived configuration object 16 is a subset of a configuration 10 and is chosen based on the users request. Configuration object 10 can contain multiple configurations, based on such alternative information as item views 26 and item view revisions 28.

Derived configuration object 16 is the specific, fully evaluated configuration object based on the information in a client's query.

FIGURE 4 illustrates the application programing interface (API) for PSCM system 60 in accordance with the teachings of the present invention. Illustrated is a Java web top 100. Java web top 100 represents one way a client may access the PSCM. Java web top 100 may be a Java-based web browser or other Java-based application capable of communication between a client and a server. Java web top 100 is connected to one or more PSCM Java beans 102. PSCM Java bean 102 comprises one or more self contained subroutines that can be combined with other Java beans 102 to form a complete Java applet or full Java application. In this case, PSCM Java beans 102 are
developed to allow Java web top 100 to interact with and interface to PSCM system 60 without having to utilize matching hardware platforms.

If the client is not using Java enabled web top 100, an alternate way of connecting with PSCM system 60 would be used. One way for windows based systems is to develop ActiveX (or similar) components from Java Beans 102 using wrappers (bridges) to facilitate access with server side 62 or develop ActiveX components directly using "C" 108 or "C++" 110 programming language to develop an interface. ActiveX is a programming method used in Microsoft windows based applications. It is based on the Component Object Model (COM) which is designed to allow programs, and the clients and servers running them, to integrate with each other. Whether a client computer 65 uses an applet written using Java beans, ActiveX components, a "C" program or a "C++" program, the client computer 65 can also utilize CORBA to communicate with the server.

Interface 70 represents the connection between client side 61 and server side 62. Connected to server side 62 are a method handler 116 and one or more methods 118. Methods 118 are routines based on sound general software engineering principles. These may include enterprise specific routines developed by an enterprise that can extend or specialize PSCM system 60. Methods 118 are executed by method handles 116.

Middleware software can be used to facilitate client and server communication. One example is CORBA 114, which stands for Common Object Request Broker Architecture and is a standard specification for middleware for interoperability for heterogeneous computing environments. It enables applications to cross the boundaries of different computing machines, operating
systems, and programming languages. Thus, a client computer can access PSCM system 60 using a variety of software, including older software, known as legacy software, as long as a CORBA interface is designed to bridge the client and server side. This eliminates the need for proprietary front end programs to access server information. Since the present invention is drawn to a cross-platform system, CORBA 114 and middleware software similar to CORBA 114 in flexibility allow PSCM system 60 to be effectively utilized by third party software.

Thus, it is apparent that there has been provided, in accordance with the present invention, product structure configuration management service system that satisfy the advantages set forth above. Although a particular embodiment has been described in detail, it should be understood that various changes, substitutions, and alterations may be apparent to those skilled in the art and may be made herein without departing from the spirit and scope of the present invention as defined by the appended claims. References to users can include a person or a computer operable to formulate requests at a client computer.
WHAT IS CLAIMED IS:

1. A product structure configuration management system comprising:
   a client computer;
   a server computer coupled to the client computer comprising:
      a repository operable to store a plurality of view objects representing at least one product;
      a handle database operable to store a plurality of handles, each handle associated with one of the plurality of view objects; and
      a product structure configuration management server operable to receive a request from a client computer, interpret the request and return a derived configuration object to the client using the handles.

2. The system of Claim 1, when the client is operable to access one or more product structure configuration management servers.

3. The system of Claim 2, wherein the client computer is operable to access a distributed file system, the distributed file system coupled to the repository.

4. The system of Claim 2, wherein the client computer is operable to access a standard query language server, the standard query language server coupled to the repository.

5. The system of Claim 1, wherein the repository is distributed across a plurality of server computers.
6. The system of Claim 1, wherein the client computer communicates with the server through the use of a middleware software.

7. The system of Claim 6, wherein the middleware software is CORBA.

8. The system of Claim 1, wherein the client computer is operable to access the product structure configuration management server using a web browser.

9. The system of Claim 1, wherein the handle database is incorporated in the repository.

10. The system of Claim 1, further comprising an update manager coupled to the product structure configuration management server and operable to execute an action based on a change to a view object.

11. The system of Claim 1, further comprising an update manager coupled to the product structure configuration management server and operable to send an electronic mail message based on changes in the handle database.

12. The system of Claim 1, wherein the handle database is further operable to store configuration rules, the configuration rules used with the handle to formulate the derived configuration object.

13. The system of Claim 1 wherein the client computer and the server computer utilize heterogenous software and hardware.
14. A method for managing information regarding a product in a client/server computer system comprising:
   storing a model of at least one product as a plurality of view objects in a repository at a server;
   associating each view object with a handle, each handle containing information about each view object;
   storing each handle in a handle database at a server;
   receiving a request for information from a client computer at a product structure configuration system server;
   deriving a plurality of configuration rules from the request; and
   returning a derived configuration object to the client computer utilizing the handles stored in the handle database in which is a portion of the repository.

15. The method of Claim 14, wherein the client is operable to access other product structure configuration management system server.

16. The method of Claim 14, further comprising the steps of:
   checking stored view objects to detect changes in a derived configuration object; and,
   using an update manager to execute an action based on the changed view objects.

17. The method of Claim 14, further comprising the steps of accessing a dedicated file server to access files stored in the repository using the client computer.
18. The method of Claim 14, further comprising the step of accessing a standard query language database stored on the repository using the client computer.

19. The method of Claim 14, wherein the client side and the server side communicate using middleware.

20. The method of Claim 19, wherein the middleware comprises COBRA software.
FIG. 1

FIG. 4
FIG. 2

FIG. 3