PATENT ASSIGNMENT

Electronic Version v1.1
Stylesheet Version v1.1

SUBMISSION TYPE:	NEW ASSIGNMENT
NATURE OF CONVEYANCE:	ASSIGNMENT

CONVEYING PARTY DATA

Name	Execution Date
Eyron Computerized Information Systems, Ltd.	07/30/1997

RECEIVING PARTY DATA

Name:	Correlate Technologies Ltd.
Street Address:	15 Bezalel Street
City:	Ramat Gan
State/Country:	ISRAEL
Postal Code:	52521

PROPERTY NUMBERS Total: 1

Property Type	Number
Application Number:	08772801

OP $40.00 08772801

CORRESPONDENCE DATA

Fax Number: (212)245-3940
Correspondence will be sent via US Mail when the fax attempt is unsuccessful.
Phone: 2122453940
Email: b.pomerance@verizon.net
Correspondent Name: Brenda Pomerance
Address Line 1: 260 West 52 St Ste 27B
Address Line 4: New York, NEW YORK 10019

ATTORNEY DOCKET NUMBER:	0338-4001
NAME OF SUBMITTER:	Brenda Pomerance

Total Attachments: 26
source=0338-4001 asset transfer eyron to correlate#page1.tif
source=0338-4001 asset transfer eyron to correlate#page2.tif
source=0338-4001 asset transfer eyron to correlate#page3.tif
source=0338-4001 asset transfer eyron to correlate#page4.tif
source=0338-4001 asset transfer eyron to correlate#page5.tif

source=0338-4001 asset transfer eyron to correlate#page6.tif
source=0338-4001 asset transfer eyron to correlate#page7.tif
source=0338-4001 asset transfer eyron to correlate#page8.tif
source=0338-4001 asset transfer eyron to correlate#page9.tif
source=0338-4001 asset transfer eyron to correlate#page10.tif
source=0338-4001 asset transfer eyron to correlate#page11.tif
source=0338-4001 asset transfer eyron to correlate#page12.tif
source=0338-4001 asset transfer eyron to correlate#page13.tif
source=0338-4001 asset transfer eyron to correlate#page14.tif
source=0338-4001 asset transfer eyron to correlate#page15.tif
source=0338-4001 asset transfer eyron to correlate#page16.tif
source=0338-4001 asset transfer eyron to correlate#page17.tif
source=0338-4001 asset transfer eyron to correlate#page18.tif
source=0338-4001 asset transfer eyron to correlate#page19.tif
source=0338-4001 asset transfer eyron to correlate#page20.tif
source=0338-4001 asset transfer eyron to correlate#page21.tif
source=0338-4001 asset transfer eyron to correlate#page22.tif
source=0338-4001 asset transfer eyron to correlate#page23.tif
source=0338-4001 asset transfer eyron to correlate#page24.tif
source=0338-4001 asset transfer eyron to correlate#page25.tif
source=0338-4001 asset transfer eyron to correlate#page26.tif

Technology Transfer and Shared Services Agreement

Technology Transfer and Shared Services Agreement (the "**Agreement**") is being made and d into on July 30, 1997 by and between **Correlate Technologies Ltd.**, an Israeli company, of ezalel Street, Ramat Gan 52521, Israel (the "**Company**") and **Eyron Computerized mation Systems Ltd.**, an Israeli company, of 15 Bezalel Street, Ramat Gan 52521, Israel ron").

EREAS, Eyron is the owner of the intellectual property rights and other rights in a patent application, software and other know how and proprietary information related to a certain computer software and user interfaces for information management, enabling entering and managing information on a computer system using a natural user interface; and

HEREAS, the Company has been incorporated by Eyron to take advantage of and commercialize the aforesaid software and user interfaces; and

HEREAS, Eyron wishes therefore to transfer to the Company all of its title, right and interest in and to the abovementioned rights, and the Company wishes to accept all such rights; and

HEREAS, the Company requires for the operation of its business certain resources, and the parties have agreed that the Company shall share with Eyron some or all of its leased premises, employees, support services, office and communication devices, cars and other scientific, technical and office equipment (collectively, "**Eyron's Resources**") on the terms and conditions agreed upon herein; and

HEREAS, the parties wish to set forth in writing the terms and conditions of their agreement.

OW, THEREFORE, in consideration of the mutual promises and covenants set forth herein, and nding to be legally bound hereby, the parties agree as follows:

1. **Preamble**

1.1. The Recitals hereto consist an integral part hereof.

1.2. The headings of the Sections and Sub-Sections of this Agreement are for convenience of reference only and are not to be considered in constructing this Agreement.

1.3. For the purposes of this Agreement, the term **"Technology Rights"** shall mean any and all rights which Eyron owns with regard to any and all patents' applications, trademarks, copyrights, service marks, trade names, moral rights, mask works, technology, know how, trade secrets and information related to the technology of computer software and user interfaces for information management, enabling entering and managing information on a computer system using a natural user interface, all as described in more detail in: (i) the document titled "Computer Software and User Interface for Information Management" attached hereto as **Exhibit 1.3A**; (ii) the document re "Patent Application for Seamless Web" attached hereto as **Exhibit 1.3B** (iii) U.S. Trademark Applications for Co-Relate, O-Paper and Open Paper attached hereto as **Exhibit 1.3C**; (iv) documents with regard to U.S. Patent Application re: "Computer Software and User Interface for Information Management" no. 08/772,801, filed on December 24, 1996, attached hereto as **Exhibit 1.3D**; (v) and including any and all rights of Eyron pursuant to an Agreement between Eyron and Theodore M. Diament, dated December 30 1996 attached hereto as **Exhibit 1.3E**, all documents, data in another tangible form and information related to the aforesaid, such as source codes, object codes, computer programs, flow charts and related documentation, in any form whatsoever, and all permits, grants and licenses or other rights running to or from Eyron relating to any of the foregoing and any and all applications for any of the foregoing.

2. Representations and Warranties of Eyron

Eyron represents and warrants to the Company that the following representations and warranties are true and accurate in all respects, and acknowledges that the Company is entering into this Agreement in reliance thereon:

2.1. To the best of Eyron's actual knowledge, Eyron is the sole owner of the entire right, title and interest in the Technology Rights. No portion of the Technology Rights has been assigned, licensed, pledged, mortgaged or conveyed, in whole or in part, to any third party.

(i) To the best of Eyron's actual knowledge, no Technology Right is subject to any law, outstanding order, stipulation or agreement restricting the use or licensing thereof; (ii) to the best of Eyron's actual knowledge, no person or entity, other than Eyron, has any ownership right, title, interest, claim in or lien on any of the Technology Rights; (iii) Eyron is not aware of any third party that is infringing or violating any of the Technology Rights; (iv) Eyron has not granted, and, to the best of Eyron's actual knowledge, there are not outstanding, any options, licenses or agreements of any kind (other then the agreement between Eyron and Theodore Diamant, dated December 30, 1996 (attached in Exhibit 4.11 to the Agreement to which this agreement is annexed) as relating to the Technology Rights; (v) Eyron has not agreed or undertaken to pay any royalties or other payments to third parties with respect to the marketing, sale, distribution, manufacture, license or use of any right relating to the Technology Rights, and, to Eyron's best actual knowledge, there is no obligation of the part of it to pay any such royalties or other payments.

2.2. To the best of Eyron's actual knowledge, the Technology Rights have not violated or infringed, are not currently violating or infringing, and Eyron has not received any communications alleging that Eyron (or any of its employees or consultants) has violated or infringed or would violate or infringe, any right of any person, by reason of the Technology Rights.

2.3. To the best of Eyron's actual knowledge, at no time during the conception of or reduction of any of the Technology Rights to practice was any developer, inventor or other contributor to such Technology Rights operating under any grants from any governmental entity or agency, performing research sponsored by any governmental entity or agency or private source, or subject to any employment agreement, or invention assignment or nondisclosure agreement, or other obligation with any third party that in each case could adversely affect Eyron's rights in such Technology Rights in any material respects.

2.4. Eyron has taken security measures to protect the confidentiality and value of all the Technology Rights, which measures are reasonable and customary in the industry in which Eyron operates.

3. Transfer of the Technology Rights

3.1. As of the date hereof, Eyron absolutely, fully and irrevocably transfers, assigns, conveys and delivers to the Company any and all of the Technology Rights, and the Company hereby acquires and takes assignment and delivery of all and any said Technology Rights.

3.2. To remove any doubt, this transfer, assignment, conveyance and delivery is on a worldwide basis and includes all and any of Eyron's rights of any type or nature whatsoever with regard to the Technology Rights, for the ownership, use and benefit of the Company, its successors, assigns and legal representatives.

3.3. Eyron, on behalf of itself and its representatives, hereby covenants and agrees, without further consideration, to do all such lawful acts and things and to execute such further lawful assignments, documents, assurances, applications and other instruments as may be required by the Company in order to cause all and any of the Technology Rights to vest in the Company, its successors, assigns and legal representatives.

3.4. Without derogating from the above provisions, Eyron shall transfer to the Company, within seven days from the date hereof, all documents, data in tangible form and information related to the Technology Rights, such as source codes, object codes, computer programs, flow charts and related materials, and shall not keep any copies, summaries, or other documents or data in tangible form containing any of the Technology Rights.

3.5. Without derogating from any of Eyron's representations, warranties and undertakings under this Agreement, in the event that any third party shall claim to have any right to any of the Technology Rights, or that such Technology Rights infringe upon any rights, all in conflict with the provisions of this Agreement, Eyron shall fully cooperate with the Company in order to assert the Company's rights in the Technology Rights pursuant to this Agreement.

4. **Transfer of Employees**

4.1. The parties agree that the employees listed in **Exhibit 4.1** are on the date hereof employees of Eyron and shall be, as of the date hereof, employees of the Company.

4.2. Eyron shall assign to the Company all Managers' Insurance policies providing for severance benefits and all other accrued benefits with respect to the employees referred to in Section 4.1 above, or with respect to any employee who will be transferred in the future to the Company, if so agreed by the parties (the "**Transferred Employees**"), including, inter alia, all savings or similar components of such policies for the benefit of such Transferred Employees or under any other undertaking of Eyron with respect to such Transferred Employees

4.3. Eyron shall indemnify the Company and hold it harmless from and against any claim, loss, damage or expense relating to amounts due to any of the Transferred Employees with respect to the period that such employees were employed by Eyron.

5. **Shared Services**

5.1. Subject to Section 5.7 below, unless the Company's requirements for services impose on Eyron an unreasonable burden and cause a material interruption or interference of or with Eyron's business, Eyron agrees to share Eyron's Resources with the Company (the "**Shared Services**"), as the Company and Eyron shall agree from time to time, in consideration of the reimbursement by the Company of Eyron's costs and expenses pursuant hereto. Eyron shall provide said Shared Services diligently, in reasonably high standard and in accordance with agreed upon time schedules.

5.2. The Company shall compensate Eyron for the provision of Shared Services on a relative basis, i.e., the Company shall reimburse Eyron for the appropriate relative part, based on actual use, of all costs and expenses incurred by Eyron for the maintenance and provision of such Shared Services.

5.3. Without derogating from the generality of the above, in respect of the following specific Shared Services, the Company shall compensate Eyron as follows:

5.3.1. Premises The Company shall compensate Eyron for the costs and expenses incurred by Eyron in respect of its occupancy of any part of the premises held presently or in the future by Eyron (the "**Premises**"), based on the relative part of the Premises occupied by the Company. With regard to parts of the Premises co-occupied by employees or equipment of both Eyron and the Company the compensation paid by the Company shall be determined according to the relative use of that part of the Premises by each of the parties, as shall be determined from time to time in writing by the parties

5.3.2. Employees. The Company shall compensate Eyron for the costs and expenses incurred by Eyron in respect of the relative number of hours dedicated by employees of Eyron to the management and business of the Company (the "**Dedicated Hours**"). The parties shall agree on an appropriate method for calculating and verifying the number of Dedicated Hours. In connection herewith, Eyron acknowledges and agrees that Messrs. Eylon Rosenstock and Ron Neumann, while they continue to be employed by Eyron, shall be also employed by the Company as Joint Chief Executive Officers of the Company, and together they will dedicate to the Company 50% of their joint business time.

5.3.3. Communication; Equipment; Supplies. The Company shall compensate Eyron for the costs and expenses incurred in respect of the Company's use of communication services, equipment and supplies on the basis of actual use and consumption by the Company. The parties shall agree on an appropriate method for recording such actual use and allocating the costs and expenses thereof.

5.4. The Company shall pay to Eyron the consideration due for the Shared Services on a monthly basis, within the first (10) ten days of each calendar month in respect of Shared Services rendered in the previous month.

5.5. For avoidance of any doubt, no Shared Services, Eyron's Resources, support or any other service rendered by Eyron to the Company shall be deemed to give Eyron any right whatsoever in the Technology Rights transferred to the Company, or in any other intellectual property right or other rights that the Company may have, relating to the Technology Rights or any other activity or business that the Company may have.

5.6. In the event that the Company shall wish to make use of services rendered by third parties in addition to any of the Shared Services, and to the extent it is not materially detrimental to Eyron, Eyron shall fully cooperate with such third parties.

5.7. With regard to the Shared Services, this Agreement shall become effective on the date hereof, and shall remain in full force and effect until terminated: (i) by the Company, when it gives Eyron written notice of termination at least 90 (ninety) days prior to the termination date; or (ii) by Eyron, when it gives the Company written notice of termination at least 90 (ninety) days prior to the termination date, provided that Eyron shall not give any such notice prior to 6 (six) months from the date hereof in the absence of the Company's consent.

Confidentiality and Non-Competition

Eyron shall keep in strict confidence, and shall not use for any purpose whatsoever, any and all information, in any form whatsoever, relating, in any way, to the Technology Rights, except information which is or shall be in the public domain not due to any act of Eyron in breach of law or agreement and except for any information the disclosure of which is required under any applicable law.

Without limiting the aforesaid, each of the parties shall hold in confidence all information obtained or received by it as a result of or in connection with the performance of the Shared Services which relates to the other party and may be private, secret or confidential proprietary information (excluding any information the disclosure of which is required under any applicable law) (the "**Proprietary Information**"). Each of the parties hereto further agrees not to disclose or use in an unauthorized manner any Proprietary Information. The parties hereto acknowledge and agree that all such Proprietary Information, not otherwise known to the public, is confidential and proprietary and shall not be disclosed to third persons without the prior written consent of the affected party.

6.2. During the term of this Agreement and for a period of 24 (twenty four) months after the latter of: (i) the provisions of this Agreement with regard to the Shared Services have terminated, as per Section 5.7 above; or (ii) the Company is no longer under the control of Eyron, Eyron shall not engage in any activity, including research, development, manufacture, sale, marketing or consulting, which relates to technologies or products which compete with technologies or products of the Company.

6.3. Eyron shall ensure that any of its present and future employees, consultants or contractors that have access to the Technology Rights shall execute a confidentiality and non competition undertaking similar to the provisions of Sections 6.1 and 6.2.

Miscellaneous

7.1. This Agreement embodies the entire agreement between the parties hereto and supersedes all other agreements or understandings between the parties in connection with the subject matter hereof. Neither party shall be bound by or liable to the other for any representation, promise or inducement made by any agent or person in the other's employ not embodied in this Agreement.

7.2. This Agreement cannot be amended, modified or supplemented except by a writing signed by all parties hereto prior to such change.

7.3. The laws of the State of Israel, without giving effect to conflict of law rules, shall govern the interpretation and enforcement of this Agreement. Any dispute arising under or with respect to this Agreement shall be resolved exclusively in the competent court in Tel Aviv, Israel.

7.4. If any provision of this Agreement is held by a court of competent jurisdiction to be unenforceable under applicable law, then such provision shall be excluded from this Agreement and the remainder of this Agreement shall be interpreted as if such provision were so excluded and shall be enforceable in accordance with its terms; provided, however, that in such event this Agreement shall be interpreted so as to give effect, to the greatest extent consistent with and permitted by applicable law, to the meaning and intention of the excluded provision as determined by such court of competent jurisdiction.

EDOC

7.5. No delay or omission to exercise any right, power, or remedy accruing to any party upon any breach or default under this Agreement, shall be deemed a waiver of any other breach or default therefore or thereafter occurring. Any waiver, permit, consent, or approval of any kind or character on the part of any party of any breach or default under this Agreement, or any waiver on the part of any party of any provisions or conditions of this Agreement, must be in writing and shall be effective only to the extent specifically set forth in such writing.

7.6. Each of the parties shall not assign, transfer, mortgage or otherwise dispose of any of its rights and/or obligations hereunder, in whole or in part, except with the other party's prior written consent. Any assignment without prior written consent shall be void.

7.7. Any notice under this Agreement shall be in writing and shall be deemed to have been duly given for all purposes (a) when received or seven (7) days after it is mailed by prepaid registered mail; (b) upon the transmittal thereof by facsimile; or (c) upon the manual delivery thereof, to the respective addressee or fax numbers set forth above or to such other address of which notice as aforesaid is actually received.

WITNESS WHEREOF, the parties hereto executed this Agreement on the date first written

ATE TECHNOLOGIES Ltd.

Technologies Ltd.

איקון מערכות בע"מ
ממוחשבת בע"מ

Eyron Computerized Information Systems Ltd.

doc

EXHIBIT 1.36

APPENDIX A2

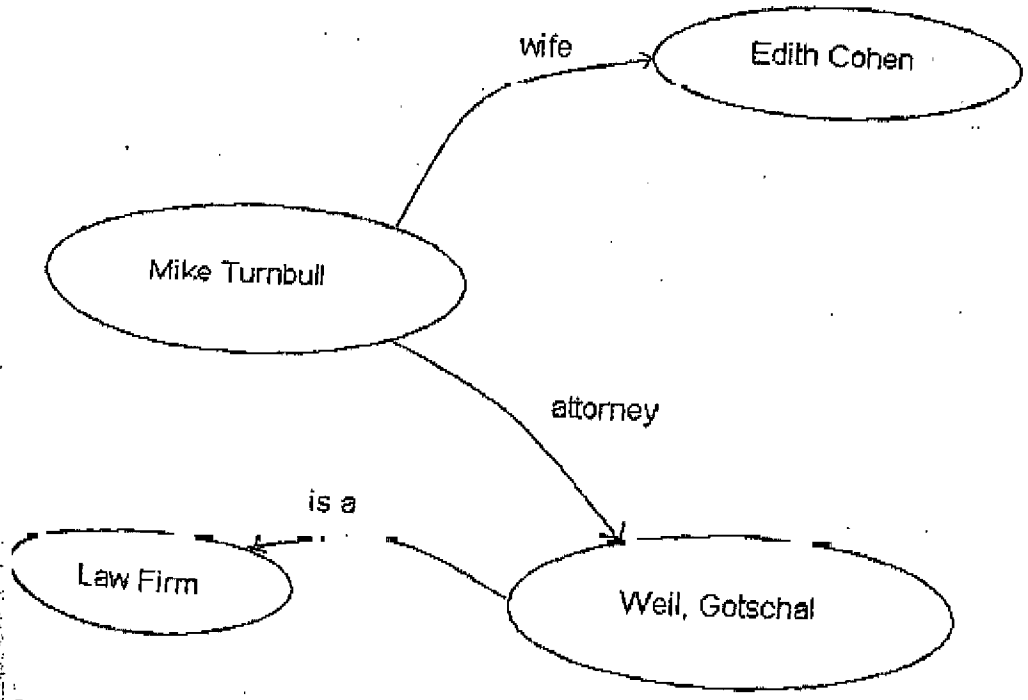
CONFIDENTIAL ATTORNEY-CLIENT COMMUNICATION

From: Ted Diament on Behalf of Eyron Information Systems Ltd.
To: Michael Dierks, Weil, Gotshal & Magnes, LLP
Re: Patent Application for Seamless Web
Date: November 25, 1996

Seamless Web ("Web") is a software application enabling the efficient input, storage, manipulation, extraction and analysis of loosely structured and unstructured data. Unlike database systems which require that data first be organized in tables, records and fields, Web allows the user to enter data without contemplating an overall structure and without forcing the data to conform to predetermined categories. Data is captured as it is discovered and generated in the course of business, and a structure for the data emerges naturally as an incident of data entry. Web brings the power of database searching and analysis to domains where the nature and scope of the information generated may be unpredictable in advance of its discovery or production.

Web is based on a variant of the concept of the semantic network. A semantic network is a knowledge representation paradigm that has been the subject of research in Artificial Intelligence since the 1960's. The central idea is that knowledge consists of objects and their relationships to each other. The objects are represented by nodes and the relationships are represented by labeled arcs between the nodes. A traditional semantic network might be rendered as follows:

Diagram Traditional Semantic Network



30, 1996
אירון מע
ממוחשב

001

EYRON

03/11 2005 18:[illegible] 7752649

network is intended to express the knowledge that Mike Turnbull is an attorney at the law firm called Weil, Gotschal and that his spouse is Edith Cohen. The semantic network shown above is obviously very simple. As more nodes and arcs are added, the network becomes increasingly unwieldy to draw. Traditionally, semantic networks are represented only in internal computer data structures. This internal structure can then be queried in a variety of ways to extract information or to deduce logical inferences. Graphical rendering of the network is not deemed an essential feature of the traditional systems.

Web represents an advance beyond existing technology in the following respects:

1. Novel Graphical Interface for Constructing Semantic Networks

Web presents a novel graphical user interface for constructing the semantic network. The interface represents an advance over the prior art for inputting such networks.

(a) Creation of objects from a toolbox.

(b) Positioning of objects anywhere on the screen

(c) Linking objects graphically.

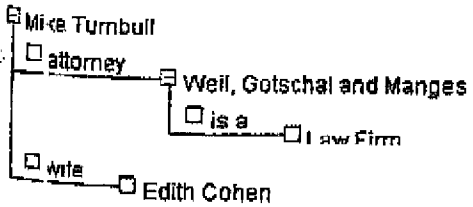
(d) In-place editing of objects

(e) Layout Control allows semantic networks to be input within semantic networks.

2. Novel Presentation Formats for Semantic Networks

Web is capable of rendering the semantic network in a variety of presentation formats which represent an advance over the prior art for displaying such networks. The basic format is shown below

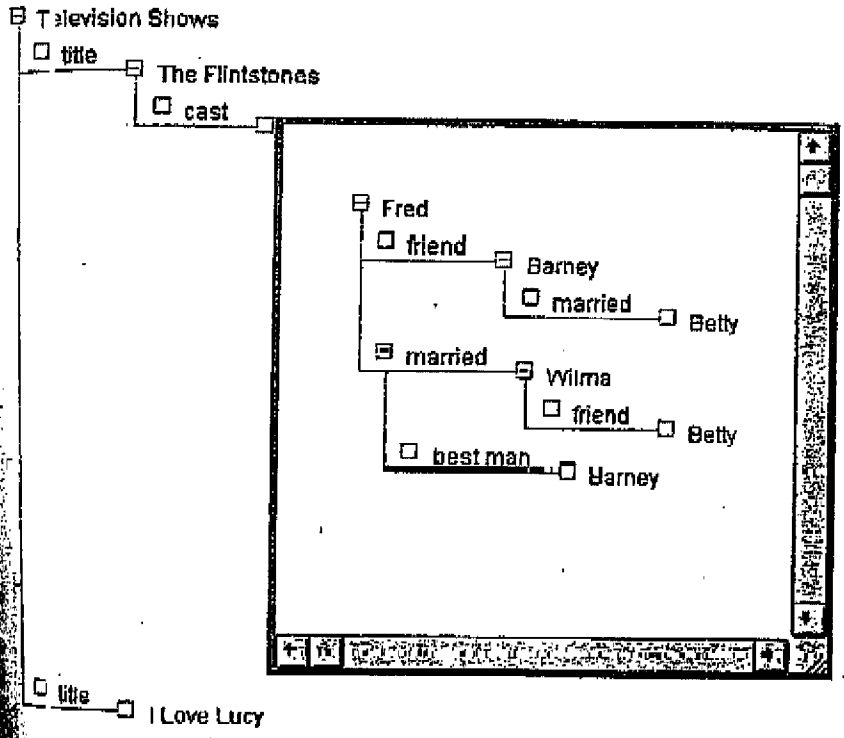
Diagram: Traditional network as displayed by Web.



(a) Web enables the user to simultaneously view distinct portions of the network.
(b) Web enables the user to simultaneously view different portions of the network in distinct formats.
(c) Web enables the objects in the network to be displayed in a format known as "in-place active." This means that objects representing data created from diverse computer applications (e.g., blueprints, medical images, scientific equations, etc) can all be displayed within the graphical framework for display of the semantic network. The objects can also be directly manipulated and modified in place without leaving the Web framework.

(d) Among the objects that can be displayed "in-place active" are themselves Seamless Web objects. The example below shows a Web within a Web

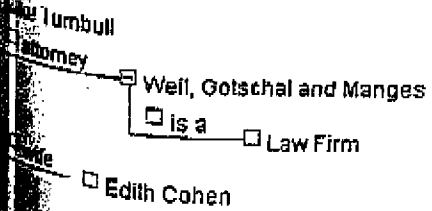
Diagram: A Web within a Web



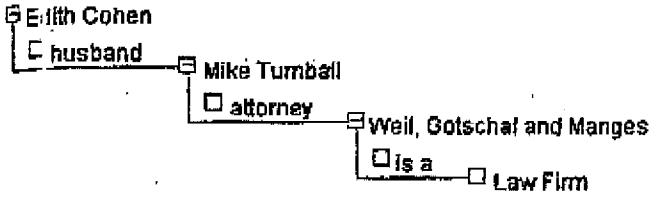
Novel Graphical Method for Exploration of Network

...troduces a novel graphical user interface which enables the user to explore the semantic network in ...which represent an advance over the prior art for exploring such networks. These include

...ploring the network from different perspectives: The user can drag and drop any node in the ...twork to an empty region on the display and explore the network from the perspective of that node. ...is operation is facilitated by the fact that Web stores information relating to the semantics of the ...bels on the arcs. For example, if the user creates the network shown below,

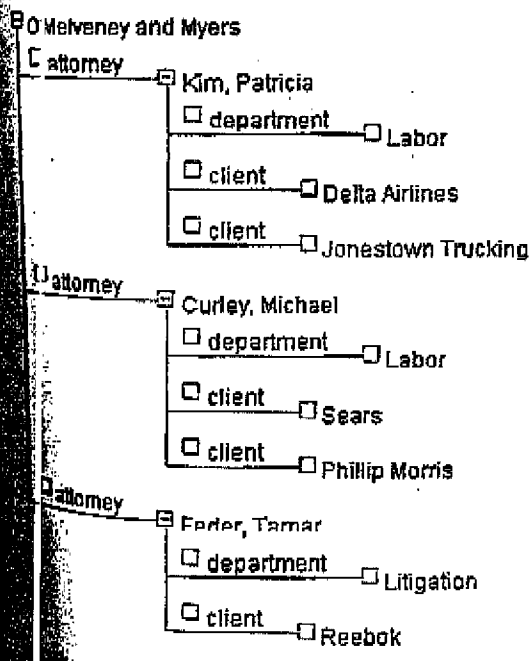


Then when viewed from the perspective of the node labeled "Edith Cohen" the network will appear as follows:

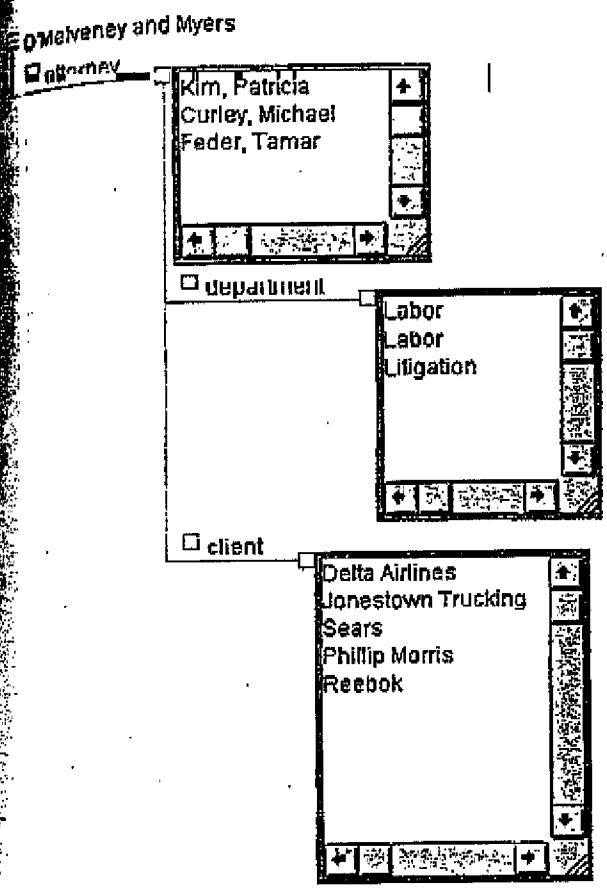


Note that the arc labeled "wife" has changed its label into "husband" it is, however, the same arc and was inputted only once. Web has built-in knowledge that if A is the wife of B then B is the Husband of A. This knowledge concerning the semantics of link labels can be augmented and overridden by the user.

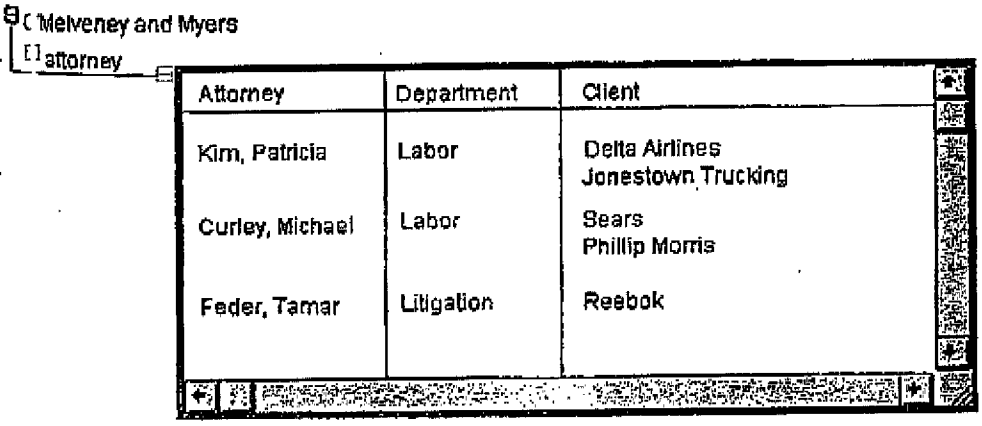
(b) Complex semantic networks can be instantly converted to "grouped format" thus the following network,



automatically reformatted and viewed with the arcs and objects grouped as follows:



(c) Any portion of the semantic network can also be viewed in chart format. For example,

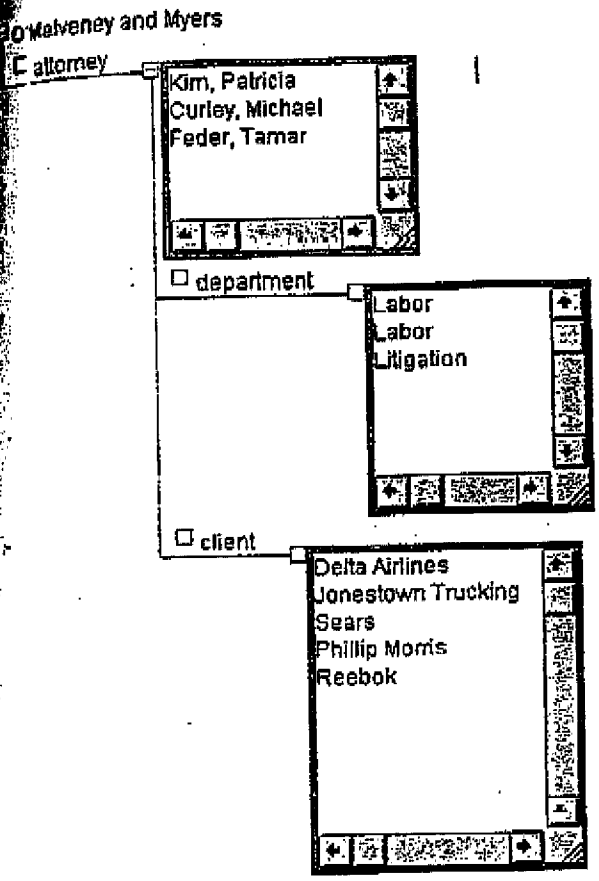


4. Incorporation of Semantic Information

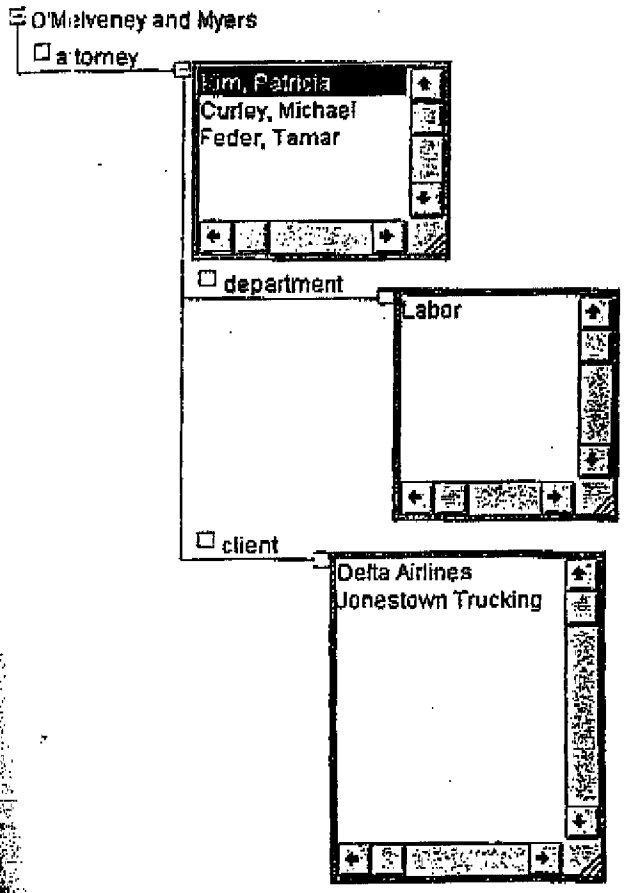
Web incorporates semantic information about the meaning of the labels on the arcs (see item 3a where this feature is presented as a user interface feature).

Novel Graphical Interface for Search and Analysis

enables the user to employ graphical means to extract information and analyze the data stored in the rk. This graphical approach represents an advance over traditional methods of querying such rks with rarefied formal query languages. For example if the web in "Grouped" format, the user can glance, view all attorneys at O'Melveney and Myers and all their clients.

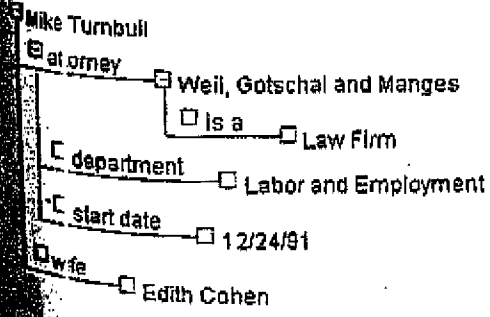


If a particular entry in one of the list boxes is selected, or several entries are multiply selected, the other list boxes automatically display only those arcs and links related to the selected items. Thus in the example below it is possible to determine, with a single click of the mouse, which department attorney Patricia Kim is in and who her clients are.



6. Extended Conception of Semantic Network

Web extends the concept of the semantic network in a new direction by conceiving of the labeled arcs, themselves, as objects. Thus, in Web, the labeled arcs, and not merely the nodes, can have arcs emanating from them. This enhances the expressive power of the network. For example, in Web, the following network can be created:



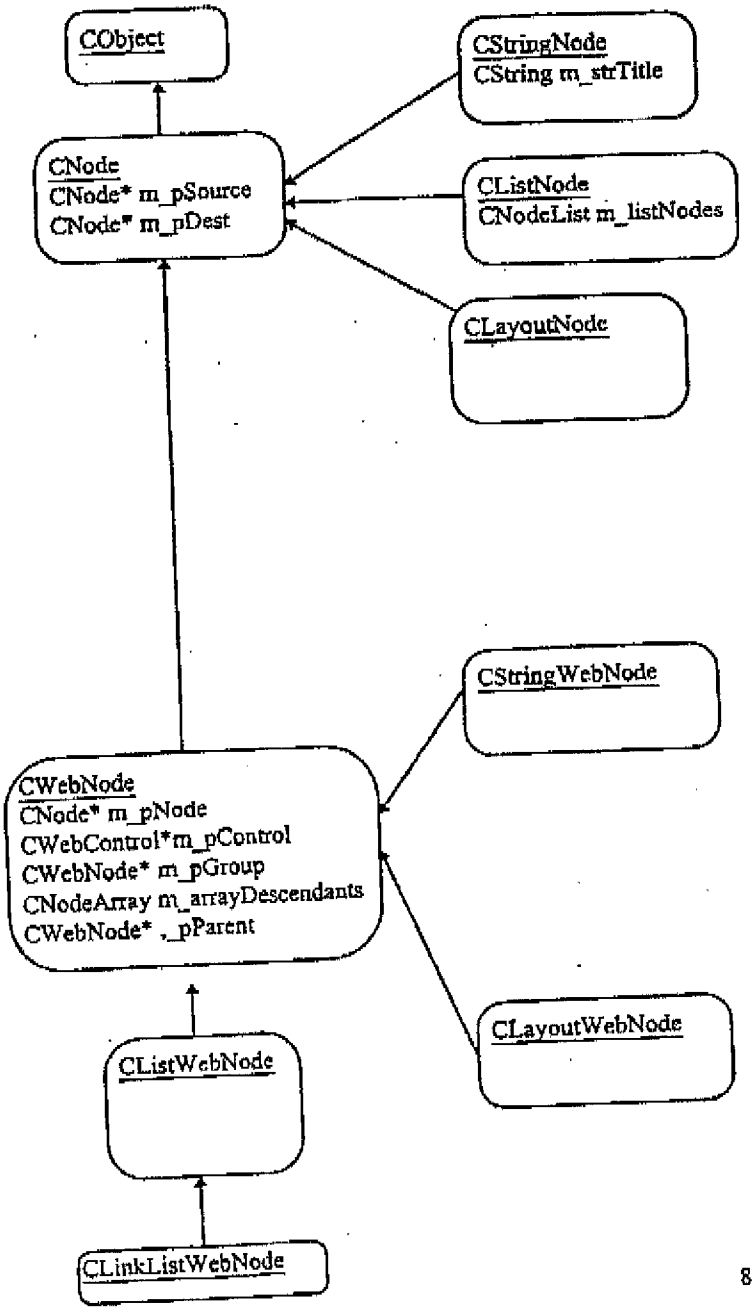
network is intended to express the knowledge that Mike Turnbull is in the Labor and Employment ment at Weil Gotschall and that he began working there on December 24, 1991. While the same

ncept could be represented in a traditional semantic network it might require a comprehensive structuring of the network and possibly a new conception of the fundamental objects and relationships ng represented. By allowing arcs off arcs, Web allows the additional information to be added to the work in a more continuous fashion.

Novel Data Structures

eb incorporates novel data structures. Web is implemented in an object-oriented fashion in C++. The s hierarchy is organized as follows:

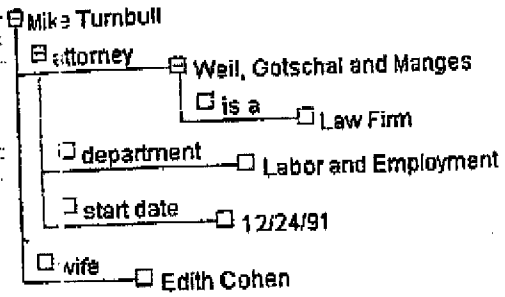
Diagram: Booch Diagram of class hierarchy. The most important classes for our purposes here are CNode, CListNode CWebNode, CListWebNode and CLinkListWebNode.



From this class structure it is possible to discern some fundamental novelties of the data structure which represents an advance over the prior art for storing semantic networks.

(a) Arcs are themselves represented as semantic network objects from which other arcs can emanate.

Traditional semantic networks consist of objects and the arcs between them. In Web, both arcs and objects are represented with the same fundamental data type -- CNode. An object in Web's version of the semantic network is represented as a CNode in which both member variables CNode::m_pSource and CNode::m_pDest are set to NULL. An arc is represented as a CNode in which the member variable CNode::m_pSource points to the object from which the arc emanates and the member variable CNode::m_pDest points to the object the arc is directed at. Thus the arc is represented as a CNode and the source and destination of the arc are stored as pointers in the member variables CNode::m_pSource and CNode::m_pDest. Both of these are declared as CNode pointers and thus can indicate any CNode including both objects and arcs. Thus in Web, unlike in the standard conception of semantic networks, an arc is also an object. An arc can therefore have other arcs emanating from it as is illustrated by the semantic network shown below:

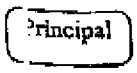


(b) Information concerning the display and rendering of the semantic network is itself stored as a semantic network. This technique greatly simplifies implementation of the application.

The CWebNode data type is designed to represent information pertaining to the display and rendering of a portion of the semantic network. But the CWebNode class derives from CNode, i.e., it is a CNode in all respects, and thus is itself part of the data of a semantic network.

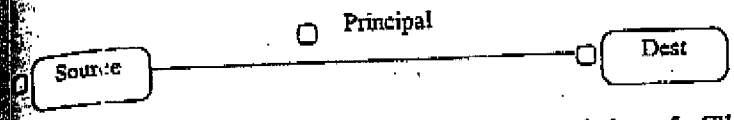
You might think of a CWebNode as a CNode on display. The CNode contains the underlying data pertaining to the semantic network arc or object, the CWebNode contains information concerning the rendering of that particular CNode on the display. Thus a CWebNode contains a pointer CWebNode::m_pNode to the underlying CNode being displayed. It also holds a pointer to a CWebControl which is the abstract class responsible for rendering the data represented by the CNode. A CWebNode also maintains a list of descendants all of which are CWebNodes. But for our purposes here, the most important point is that a CWebNode is itself a CNode. Thus it has an m_pSource and an m_pDest pointer which can point to other CNodes, and in this both m_pSource and m_pDest will point to CWebNodes. A CWebNode exploits the data structure of the CNode to store information concerning the manner in which an arc or an object is rendered on the display. The manner in which this is achieved is as follows.

A WebNode with standard style, no source, no dest and no displayed descendants would display as follows:



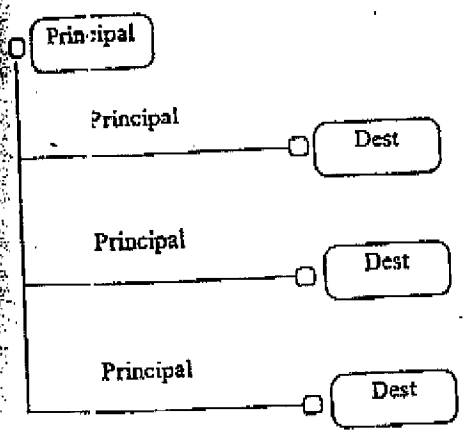
A WebNode with standard style dest WebNode but with no source WebNode is displayed as follows:

Finally, if the CWebNode has both source and dest pointers it is displayed as:

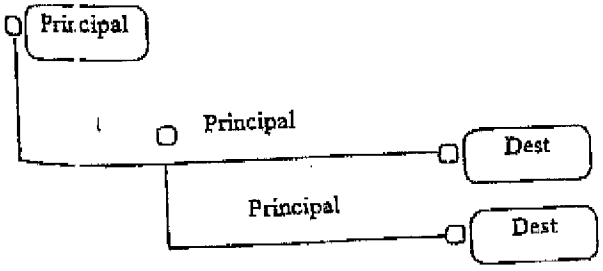


The three diagrams above represent the three standard renderings of a CWebNode. Which rendering is used depends on whether the CWebNode has a source a dest, both or neither.

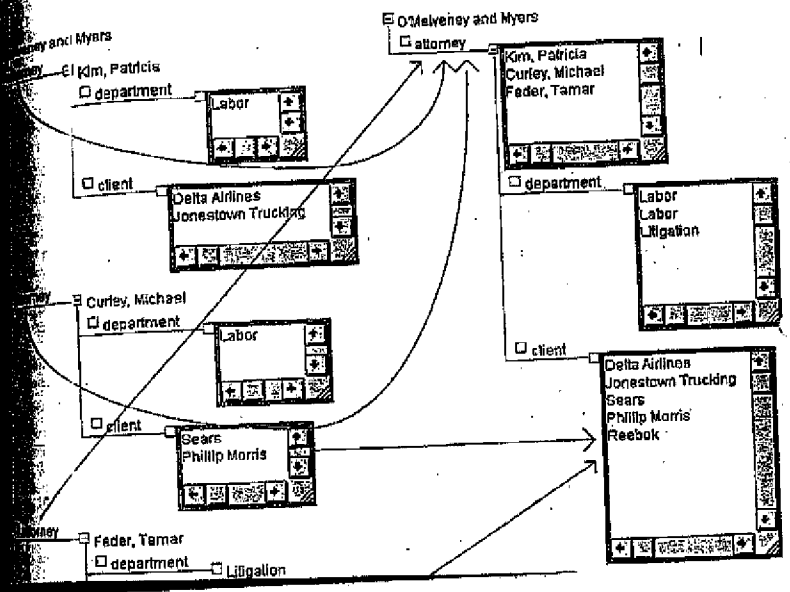
A CWebNode, is a CNode but it extends that data struture by maintaining a list of descendants. If we take one of the three standard CWebNode forms and add a list of descendant CWebNodes we can create more complicated renderings. For instance, if we take the standard style CWebNode with no source and dest and provide it with a descendant list of CWebNodes each with a dest CWebNode but no source, the CWebNode is displayed as follows



Things get interesting when these descendant WebNodes themselves have descendants.



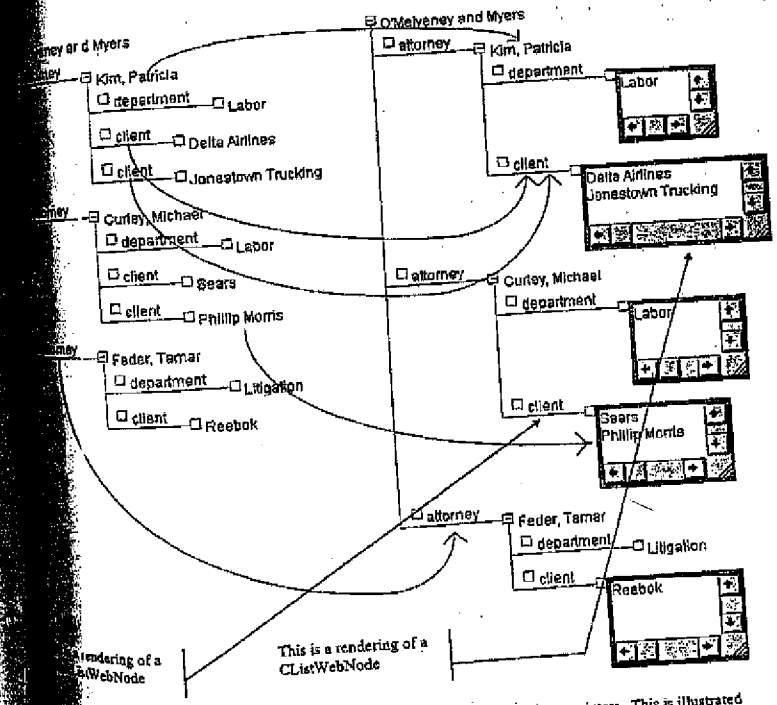
The algorithm for displaying a CWebNode is described completely in the section on algorithms below. For our purposes here we note that because of the clever design of the CWebNode data structure, the essence of



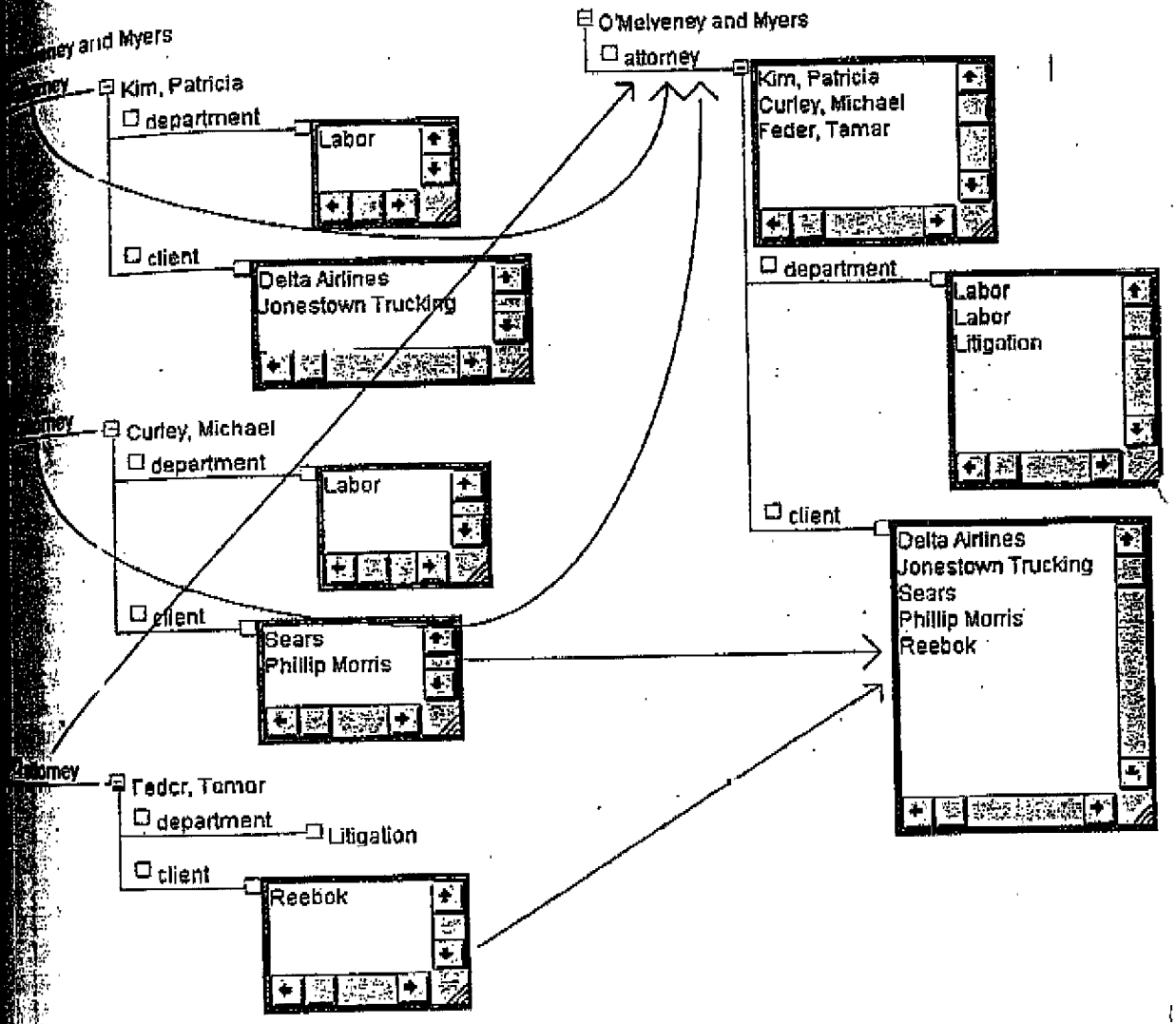
algorithm is very straightforward. In object-oriented fashion, a CWebNode is equipped with a method enables it to render itself on the display. Complex structures like the ones shown in the two previous grams are rendered by having the CWebNode render itself in one of the three standard forms and then ing it recursively request that each CWebNode in its descendant list render itself.

In addition to the nodes and arcs, web maintains internal data structures that express higher level relationships between the data items so that they can be regrouped in meaningful ways.

CWebNode holds a pointer called m_pGroup which points to its group CWebNode. This group ebNode is of one of two types CListWebNode and CLinkListWebNode. The diagram below shows the ebNodes on the left and the group CWebNodes on the right. Some of the m_pGroup pointers are ated.



even the grouped CWebNodes in the diagram on the right have m_pGroup pointers. This is illustrated he diagram below.



Now typically only one of these structures is rendered on the display, but all of them are maintained and all are updated as each new arc and object is added (the section on algorithms below discusses how the m_pGroup pointers are updated). The advantage of maintaining this data structure is that it makes it possible to switch immediately between the grouped and ungrouped renderings of the semantic network. More importantly, for our purposes here, the grouped structures are critical in the implementation of several of the algorithms essential to the application – in particular, those data structures are critical for rendering a portion of the semantic network in chart form.

8. Novel Algorithms

Web implements novel algorithms:

(a) To display the semantic network

We render the semantic network by invoking the method CWebNode::Draw passing it a parameter rcBounds which indicates the rectangular region in which it is to restrict its rendering. The full signature of the method in C++ is

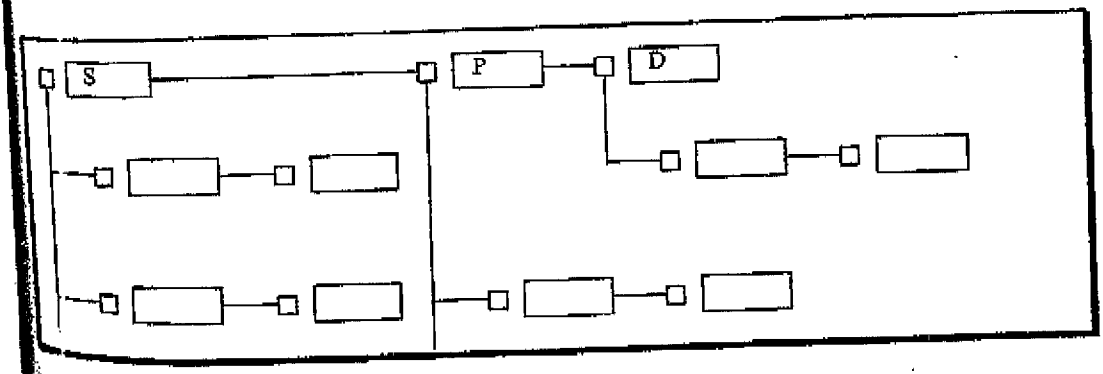
```
void CWebNode::Draw(DWORD dwAspect, CDC* pDC, const CRect& rcBounds, const CRect& rcInvalid)
```

The first of these, dwAspect, is simply a value that allows us to alter the style of the rendering. The parameter pDC is a pointer to a device context, a concept specific to the Microsoft Windows operating system that represents the device on which the CWebNode is to be rendered (printer, monitor, etc.). The last parameter, rcInvalid indicates the rectangular region within rcBounds that need to be rendered. It allows the caller to optimize the speed of the rendering by indicating the smallest region that needs to be redrawn. These parameters dwAspect, pDC and rcInvalid are not at all essential to an understanding of algorithm. We indicate them here for the sake of completeness. Only the parameter rcBounds is essential to the logic of the drawing algorithm.

This Draw method has a close relationship with the method to the method CWebNode::GetExtent which determines the dimensions required to render the CWebNode in full. Its signature in C++ is as follows

```
void CWebNode::GetExtent(DWORD dwAspect, CSize& size)
```

To understand these two methods, consider the general form of the rendering of a CWebNode shown below. The principal CWebNode being rendered is marked with a "P". The source CWebNode (the CWebNode indicated by the principal CWebNode's m_pSource pointer) is marked with an "S" and the destination CWebNode (the CWebNode indicated by the principal CWebNode's m_pDest pointer) is marked with a "D".



The key points to recognize are:

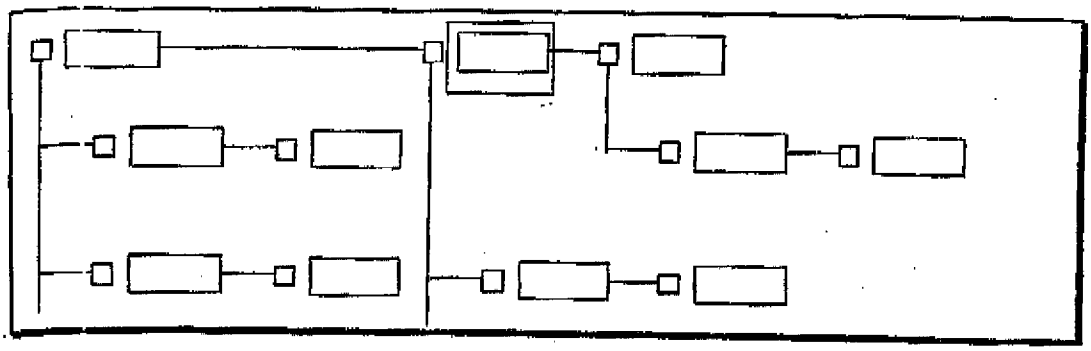
1. The descendants of the destination CWebNode push down the descendants of the principal CWebNode.

2. The descendants of the source CWebNode push the principal CWebNode out to the right.

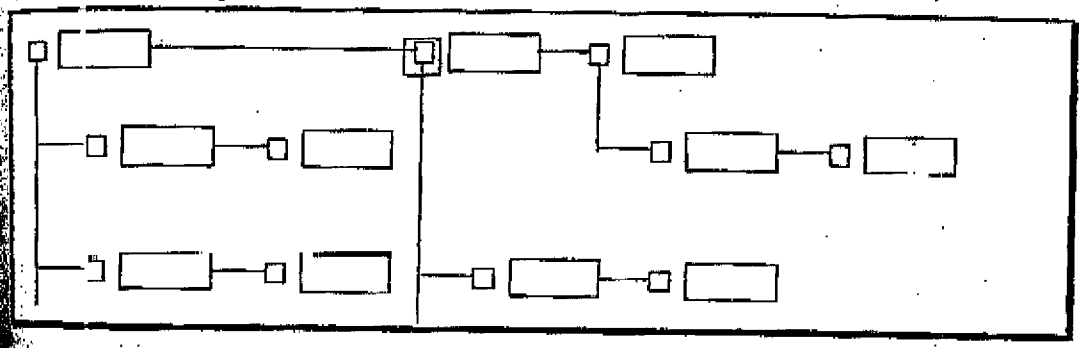
The influence of the source and dest CWebNodes on the position of the principal CWebNode and its descendants is asymmetric, but it maintains the following invariant: The position of a CWebNode does not change when its descendants are exposed or hidden.

To render the CWebNode we calculate the following rectangles:

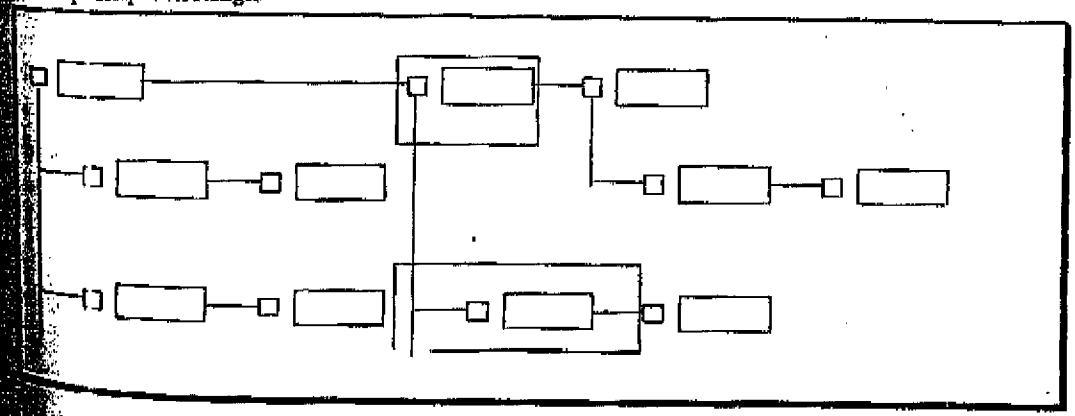
1. The Control rectangle



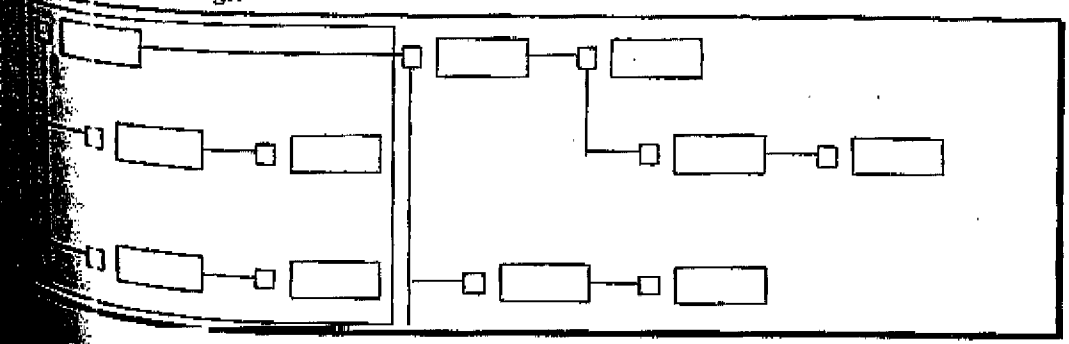
2. The Button rectangle



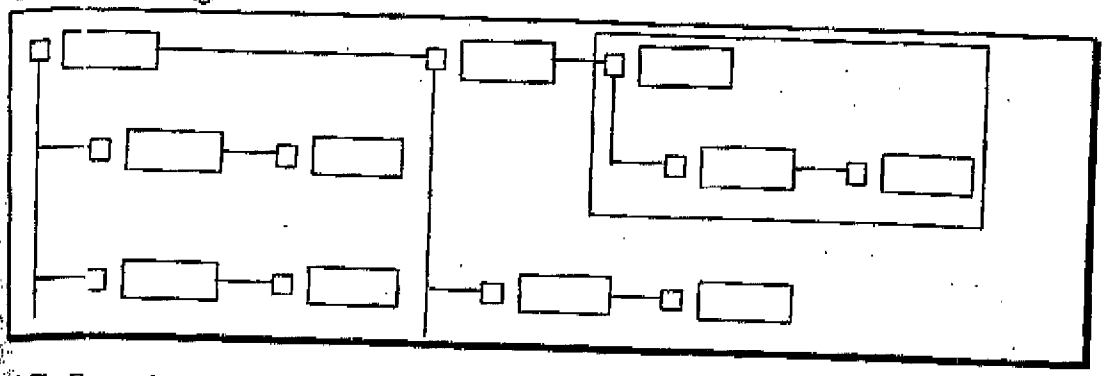
2. The principal Rectangle



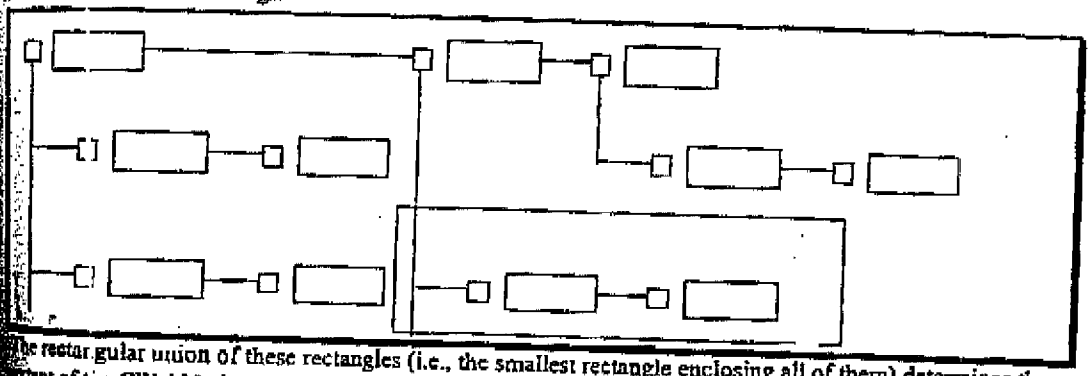
The Source Rectangle



3. The Dest rectangle



4. The Descendant rectangle



he rectangular union of these rectangles (i.e., the smallest rectangle enclosing all of them) determines the ent of the CWebNode. The CWebNode can be drawn as a principal, a source, a dest, or a descendant. hus we define another function

```
CWebNode::Draw(DWORD dwAspect, CDC* pDC, const CRect& reBounds, const CRect& reInvalid, int nWebDrawType)
```

he only thing our old Draw function does is to call this one

```
CWebNode::Draw(DWORD dwAspect, CDC* pDC, const CRect& reBounds, const CRect& reInvalid)
    Draw(dwAspect, pDC, reBounds, reInvalid, WDT_PRINCIPAL);
```

Extended function calls a procedure to calculate the four rectangles rcPrincipal, rcSource, rcDest, and rcDescendants. Then it to draw. It calls the extended Draw function on its source

```
->Draw(dwAspect, pDC, rcSource, rcInvalid, WDT_SOURCE);
```

s, of course, a recursive call (with a different "this" pointer). Next comes the guts where the control is displayed

```
ntrol(dwAspect, pDC, rcControl, rcInvalid);
```

Now draw the dest and the descendants

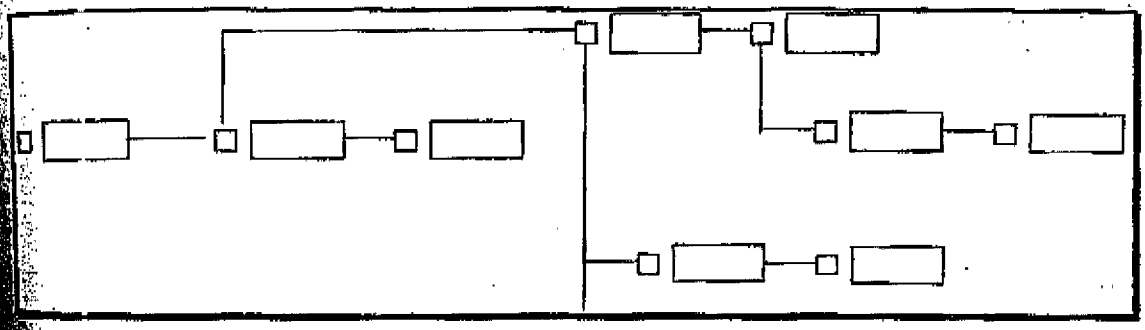
```
Draw(dwAspect, pDC, rcDest, rcInvalid, WDT_DEST);
endants(dwAspect, pDC, rcDescendants, rcInvalid);
```

DrawDescendants method simply loops through the descendants of the CWebNode calling the ed Draw function for each one.

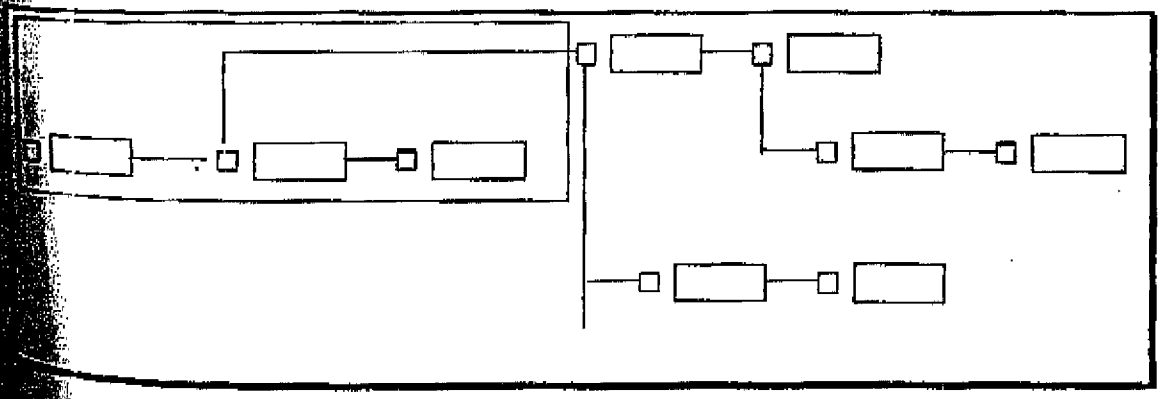
plete the extended Draw procedure we need to draw the lines connecting the controls. This is e extra parameter nWebDrawType comes into play. If the CWebNode is being drawn as

WDT_SOURCE, it is responsible for drawing the line that extends from the right end of its control to outer reaches of its rcBounds rectangle (which, of course, will be the rcSource rectangle calculated by principal CWebNode). If the WebNode is being drawn as WDT_DEST, it is responsible for drawing the little line from the left end of its control rectangle to the left end of rcBounds. For WDT_DESCENDANT, draw the vertical line along the left edge of its rcBounds rectangle as well as the short horizontal line from the left side of its control back out to the left edge of the rcBounds rectangle. Every CWebNode, regardless of the DrawType, is responsible for drawing the vertical line from the left side of its principal rectangle down to the top of the rcDecendnats rectangle.

The responsibility for drawing the lines connecting the principal WebNode with the source WebNode is assigned to the source WebNode. The source WebNode holds most of the data necessary to determine the shape of this line. For instance, the source WebNode knows whether it, itself, has both a source and a dest WebNode.[1] If it does, the line must be shaped like this:



In such a case, the Source rectangle would be



(b) To manipulate its data structures. The algorithms are capable of maintaining and updating the data structures incrementally with the addition of each new arc and object. Thus the user can request that the information in the network be reorganized, regrouped, or presented in a different

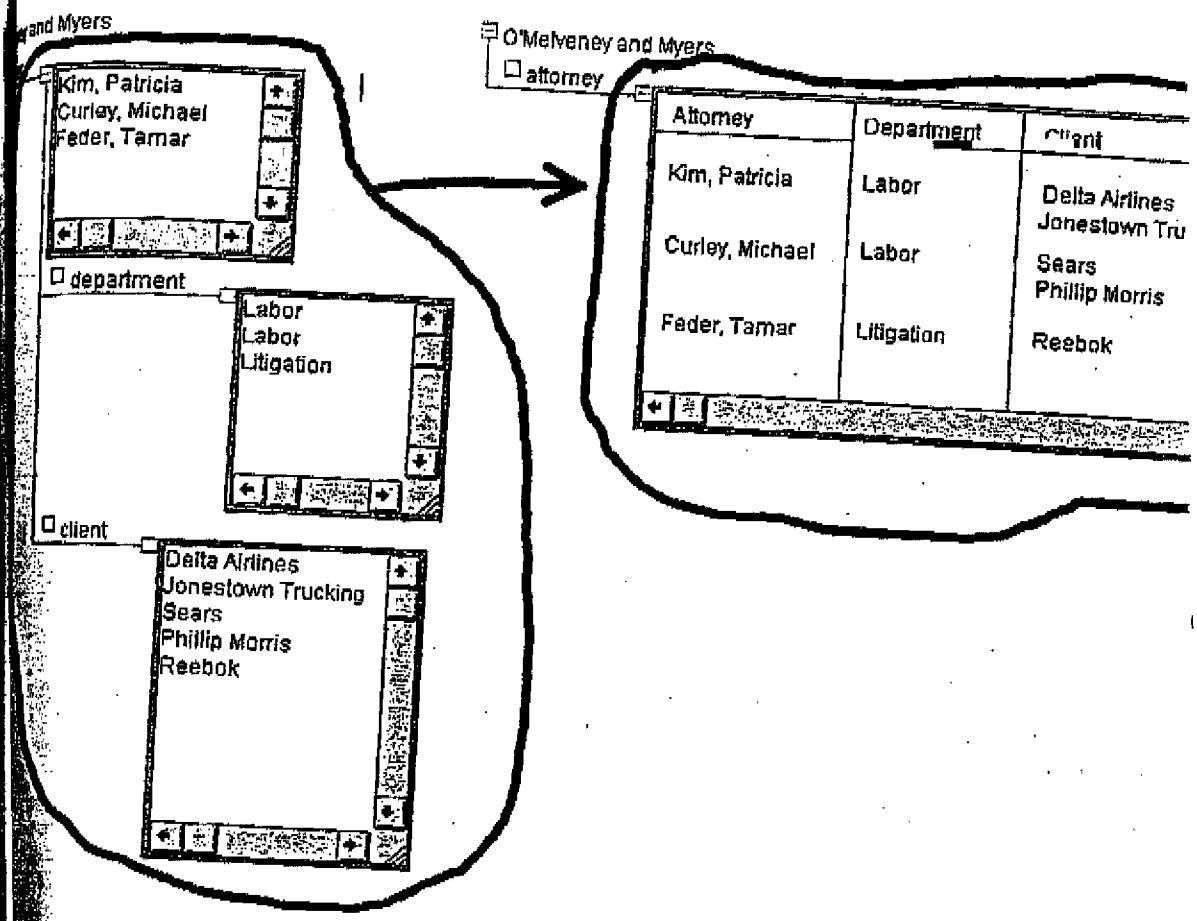
[1] ...rnative is to have the principal CWebNode be responsible for drawing lines. The principal ...ode would have to poll its source and dest CWebNodes to determine what type they are (i.e., do ...e both source and dest). The principal CWebNode would also have to poll its source and dest and ...nt CWebNodes to determine dimensions of their button rectangle and the position of the button ...relative to their upper left corner. The problem with implementing CWebNode methods that ...se results is that they require a full-blown recursive calculation to answer.

format and these operations can be performed almost instantaneously without the need for a prolonged search procedure.

When a new arc is added, emanating from a semantic network object or from another arc, in our user interface this operation corresponds to the addition of a new descendant CWebNode to the list of descendants of the CWebNode representing the object or arc from which the new arc emanates. When a new descendant is added, the m_pGroup pointers must be updated. This is achieved in the following manner

1. The new descendant CWebNode looks to its parent CWebNode's group (i.e., m_pParent->m_pGroup). It places this value in a local variable, say pParentGroup. If pParentGroup is NULL, we set pParentGroup to be m_pParent.
2. Our new CWebNode invokes a method on the CWebNode referenced by pParentGroup to determine if its own label matches the label of one of the (other) descendants of that CWebNode.
3. If there is no match we dynamically create a new CLinkListWebNode as the group node to which our descendant CWebNode will belong. We add this CLinkListWebNode as a new descendant of the Parent's Group CWebNode and this causes a recursive invocation of this same procedure at higher levels of grouping (see the discussion of grouping in the section on data structures above).
4. If there is a match then we set the m_pGroup member variable of the new descendant CWebNode to point to the matching CWebNode found in step 2.

(c) To render a portion of the network as a chart.



The CWebNode::DrawChart method takes a parameter rcBounds which indicates the rectangular region on the display in which it is to restrict its rendering. The method performs the following steps.

It calls DrawChartTitles passing the value rcBounds

It increases the coordinates of the top of the rectangle rcBounds so that the top of the rectangle lies below the line of titles just rendered.

It calls DrawChartEntries passing the value of rcBounds.

The method CWebNode::DrawChartTitles is passed a parameter rcBounds which indicates the rectangular region in which it is to restrict its rendering. It is executed as follows.

Initialize a local variable, say rcColumn, representing a rectangle to the trivial rectangle with the topleft point set to (0,0) and the width and height set to (0, 0).

If this CWebNode has an m_pDest that is non-null, skip to step 5.

Call the method GetExtent() to calculate the rectangular region required to display the column of entries in this CWebNode. Calculate the intersection of the returned value with the rectangle rcBounds. Store the resulted value in a local variable, say rcColumn.

If the m_pParent pointer is non-null, call the method DrawText on the CWebNode object pointed to by m_pParent. Pass that method the value rcColumn to indicate the rectangle in which in which it can render itself (this value was calculated in the call to GetChartExtent in the previous step). Note: The

m_pParent pointer will point to a CWebNode rendering an arc directed at this CWebNode. The label on that arc forms the title of the column in the chart in which the entries for this CWebNode will be displayed.

5. For each descendant of this CWebNode call DrawChartTitles passing that method the rectangle rcBounds but with the left side of the rectangle shifted to the right by an amount equal to the width of rcColumn.
6. If the value of m_pDest is NULL then jump to step 8. Otherwise call a method GetChartExtent to calculate the rectangular region required to display, in chart form, the entries in this CWebNode together with the entries in all its descendants. Store the value in a local variable say rcChartExtent.
7. Now m_pDest points to a CWebNode. Call the DrawChartTitles method on that CWebNode passing it the value of rcBounds but with the left side of that rectangle shifted right by an amount equal to the width of the rectangle rcChartExtent calculated in the previous step.
8. Return to the caller.

The method CWebNode::DrawChartEntries has essentially the same structure as the method CWebNode::DrawChartTitles. It, too, is passed a parameter rcBounds indicating the rectangular region in which it is to restrict its rendering. It is executed as follows.

1. Initialize a local variable, say rcColumn, representing a rectangle to the trivial rectangle with the topleft point set to (0,0) and the width and height set to (0, 0).
2. If this CWebNode has an m_pDest that is non-null, skip to step 5.
3. Call the method GetExtent() to calculate the rectangular region required to display the column of entries in this CWebNode. Calculate the intersection of the returned value with the rectangle rcBounds. Store the returned value in a local variable, say rcColumn.
4. If this CWebNode is not grouped (i.e., if m_bGrouped == FALSE) simply invoke the method to cause this CWebNode render its contents within the region rcBounds. If, on the other hand, this CWebNode is grouped (i.e., m_bGrouped == TRUE), then invoke the method DrawChartEntries on each of its member CWebNodes in turn. On the invocation for the first member CWebNode, pass the value of rcBounds. On the subsequent invocations for each member, pass the value of rcBounds but with the top of that rectangle adjusted downward by the height that was required to render the previous member CWebNode.
5. For each descendant of this CWebNode call DrawChartEntries passing that method the rectangle rcBounds but with the left side of the rectangle shifted to the right by an amount equal to the width of rcColumn.
6. If the value of m_pDest is NULL then jump to step 8. Otherwise call a method GetChartExtent to calculate the rectangular region required to display, in chart form, the entries in this CWebNode together with the entries in all its descendants. Store the value in a local variable say rcChartExtent.
7. Now m_pDest points to a CWebNode. Call the DrawChartEntries method on that CWebNode passing it the value of rcBounds but with the left side of that rectangle shifted right by an amount equal to the width of the rectangle rcChartExtent calculated in the previous step.
8. Return to the caller.