Mediator: an Intelligent Information System Supporting the Virtual Manufacturing Enterprise

B.R.Gaines, D.H.Norrie & A.Z.Lapsley
Knowledge Science Institute and Division of Manufacturing Engineering
University of Calgary
Calgary, Alberta, Canada T2N 1N4
gaines@cpsc.ucalgary.ca, norrie@enme.ucalgary.ca, lapsley@cpsc.ucalgary.ca

ABSTRACT

Mediator is an open architecture information and knowledge management system designed to provide a flexible technology to support the management of complex manufacturing activities throughout the product life cycle. A heterogeneous environment is assumed in which the sub-systems are geographically dispersed and involve different application packages, not necessarily designed to work together, multiple platforms, protocols and forms of user interface. The function of Mediator is to provide a knowledge support system for all those involved in the manufacturing process from requirements through design, engineering, production, to maintenance and recycling. It is designed to facilitate communication, compliance with constraints including physical restrictions and legal obligations, and to generally represent knowledge about any activity or sub-system relevant to the manufacturing process. This paper reports on a second-generation Mediator implementation based on World-Wide-Web protocols and browsers augmented by specialist helper and server applications.

1. INTRODUCTION

Modern manufacturing enterprises are no longer well-defined in their locations but comprise *virtual environments* in which related manufacturing operations may be widely distributed geographically and yet closed linked conceptually in terms of dependencies and material, information and knowledge flows [7]. In particular, the flows relating to the physical manufacturing operations are only one part of a complex flow of information and knowledge concerned with underlying requirements, the logic of design decisions, the availability of inventory, contractual obligations relating to customers, suppliers, employees and government agencies, and so on.

Information and knowledge technologies have come to play major roles in supporting and managing these flows, but the heterogeneity of the many different sub-systems involved make the total management of manufacturing a complex problem. In general, manufacturing systems tend to function reasonably well under the routine conditions for which they have been developed but become problematic when situations arise that deviate from these conditions. Under abnormal conditions the information available in the system may be irrelevant or misleading, and background knowledge that would be valuable in rectifying the situation is likely to be absent.

Mediator is an open architecture information and knowledge management system designed to provide a flexible technology to support the management of complex manufacturing environments. A heterogeneous environment is assumed in which the sub-systems are geographically dispersed and involve different application packages, not necessarily designed to work together, multiple platforms, protocols and forms of user interface. The function of Mediator is to provide a knowledge support system for the managers and system operators involved in running a virtual factory. It is designed to facilitate communication, compliance with constraints including physical restrictions and legal obligations, and to generally represent knowledge *about* any activity or sub-system relevant to the manufacturing process.

The Mediator design and prototype is one outcome of the *knowledge systematization* technical work stream of test case 7, GNOSIS, of the international Intelligent Manufacturing Systems (IMS) pre-competitive research program [10]. GNOSIS involves over 100 participants in 31 industry and university organizations in 14 countries, with the objective of developing a *post mass production manufacturing paradigm* involving *reconfigurable artifacts*.

The GNOSIS *knowledge systematization* work stream has been concerned with the modeling and management of information and knowledge flows throughout the complete product life cycle from initial needs through design, engineering, production, to reuse and recycling. This paper focuses on how some of the concepts developed in GNOSIS have been implemented in Mediator as a general knowledge systematization tool.

2. MEDIATOR FUNCTIONALITY

Mediator is designed to coordinate the overall manufacturing process over the complete product life cycle from requirements to recycling. In particular, it supports the integration of computer-based system design, production engineering and production sub-systems into a mutually collaborative framework for integrated design and manufacture. It provides facilities for recording and tracing decisions taken at each stage of the product life cycle, particularly the dependencies between knowledge, decisions, datasets, and so on.

A basic Mediator technology is an open architecture visual language tool for representing conceptual schema in a way that is comprehensible to users, and can be easily tailored to different application domains. Mediator supports semantic networks that can be used to represent rules, procedures and constraints with formal, operational semantics such that an inference system can be used to give advice and check for constraint violations. The representation is domain independent and the system can reason with legal constraints and corporate procedures as well as with design constraints. Less formal concept maps are also supported where the role of the system is primarily information retrieval rather than reasoning, and the formal and informal systems can be combined.

Procedures can be triggered through the user interface to the visual language, and the procedures can include other applications or general-purpose agent systems [19, 20] that support active integration between applications and between sites. Data translation and knowledge interchange are supported through such agents.

It is assumed that many of the systems to be integrated are geographically dispersed and have been developed separately without any coordination. The problem is one of heterogeneous integration to provide a powerful and usable combined system that:

i) Enhances the functionality of each of the individual sub-systems
ii) Offers users additional capabilities beyond those of the individual sub-systems
iii) Provides users with an overall coordination system that is readily comprehensible.

It is a fundamental design requirement that the basic Mediator technology should be minimal, modular, open-architecture, cross-platform, and easy to enhance, maintain and use. Mediator is only justified to the extent that it is resource-effective in improving the coordination of complex situations, and would be counter-productive if it introduced new problems and complexities.

3. MEDIATOR ARCHITECTURE

The Mediator architecture is a distributed client, distributed server design, in which multiple users can collaborate synchronously or asynchronously through processes running anywhere on the network. The implementation supports a heterogeneous environment in which there are multiple protocols and multiple forms of user interface.

Figure 1 shows the basic architecture. At the center is a collaborating and geographically dispersed user community. Beneath this is shown the computational infrastructure to support the collaboration.

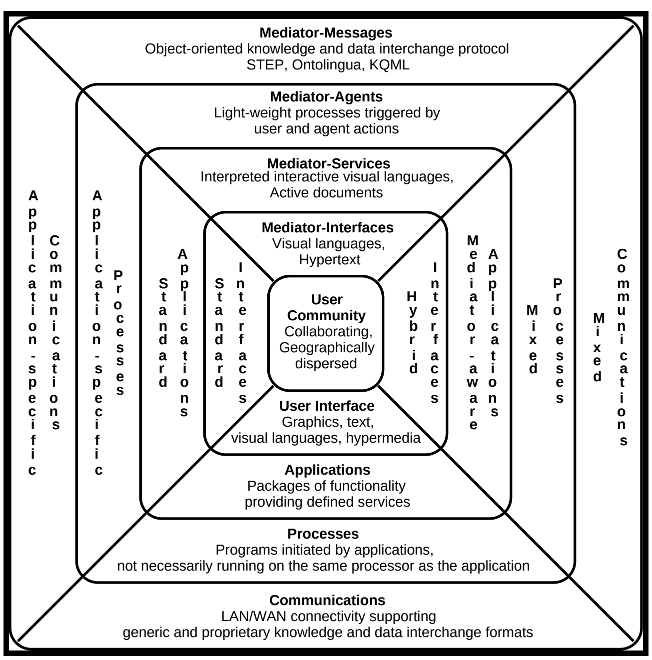

Figure 1 Mediator architecture

The community interacts with the system through a variety of forms of user interface, typically graphics, text, visual languages and hypermedia. They access a variety of 'applications' defined as packages of functionality providing defined services.

The functionality is made operational by initiating processes which may run anywhere on the network; that is, remote procedure calls are expected to be common. Inter-process communications is provided through local and wide area networks supporting a range of generic and proprietary knowledge and data interchange formats.

On the left of Figure 1, these four layers are shown instantiated in terms of conventional software packages designed independently of mediator: as standard interfaces, standard applications, application-specific processes and application-specific communications.

It is assumed that such standard applications can play a role in a Mediator-coordinated systems, minimally by the application and its datasets being registered in Mediator with its application windows open on the Mediator desktop, and maximally by Mediator controlling its inputs, outputs and operations by job-control scripts. Existing application software is assumed to play a major role in the operation of Mediator, and one can conceptualize this role by noting that the Mediator system will know a lot about such applications while they will know virtually nothing about Mediator.

At the top of Figure 1, the four layers are shown instantiated in terms of core Mediator-specific technology, what might be called a *Mediator-shell* since the majority of the software is not specific to any particular application of Mediator, but provides general collaboration and integration facilities.

The primary user interface to Mediator is through visual languages allowing the general representation of semantic systems through graphical symbols. This is supplemented by hypertext and hypermedia as appropriate. The application layer uses particular instances of these visual languages with application-specific semantics and a visual appearance designed to be natural to use in the context of the specific application.

The process layer supports agents as light-weight processes triggered by user interaction with the visual languages, and also triggered by other agents [19].

The communication layer supports object-oriented protocols for knowledge and data interchange such as STEP, Ontolingua and KQML [4]. It also supports messages in arbitrary formats as appropriate to communication with other applications not designed with Mediator in mind.

On the right of Figure 1, the four layers are shown instantiated in terms of separately designed applications that are 'Mediator-aware' to some extent, for example in using the Mediator interface technology, applications, agents or protocols as part of their normal operation.

The Mediator shell technology is designed to be highly modular and readily integrated in whole or in part with existing applications. Thus, in a design domain, one might envision a Mediator graphic interface and data interchange protocols being integrated with functional design tools such as HUT's Δ [14] or University of Tokyo's SYSFUND [15].

A detailed example of the application of Mediator agent processes to the control of manufacturing activities is given in an accompanying paper [21], and the remainder of this paper focuses on the communications and user interfaces of Mediator, and their implementation on the World-Wide Web.

4. MEDIATOR VISUAL LANGUAGE

Much of the user interaction with Mediator is through an open architecture visual language system developed to support graphic interaction with computers on a wide range of topics including knowledge representation [6, 23], concept mapping [12, 17], Petrinets [22], bond graphs [16], and so on. The system was developed as an alternative to textual interfaces to take advantage of modern graphic workstations through a simple and natural visual language of great generality. The generality is achieved because the underlying data structure is a sorted digraph which can represent typed binary relations and hence virtually any mathematical structure. A comprehensible user interface is provided through the capability to reflect the type structure in node decorations such as fonts, shading and color.

User interaction with Mediator takes place through the creation of statements in the visual language, and through interaction with these statements through popup menus whose content is specific to node type. Actions are context-sensitive: to the node selected for the popup, to nodes linked to it, and to other nodes preselected by clicking on them in the graph. This allows both simple and complex activities to be initiated by simple and comprehensible user actions. For example, Figure 2 shows a shop floor information access concept map being used to access bill of materials information for a particular product.

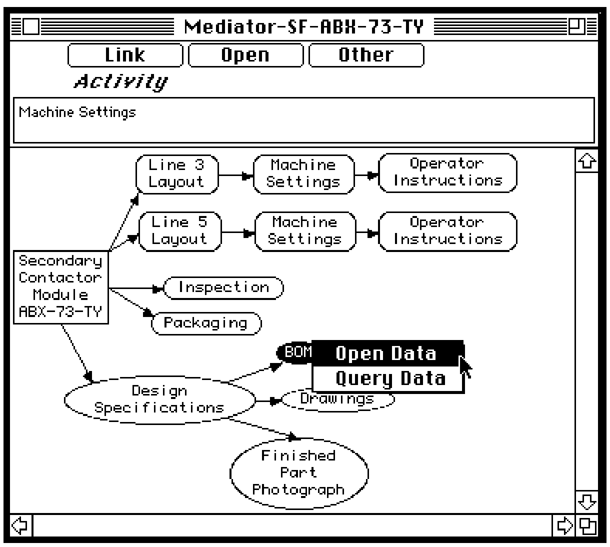

Figure 2 Interacting with a shop floor information map

The general concept map tool allows the style and functionality of such windows to be tailored to general classes of situations. Individual maps are then generated automatically or manually for particular situations, in this case a product with two possible line layouts. The user interface can be programmed using a wide range of possible dialogs. In this case a popup menu gives access to the bill of materials, through another concept map, or by initiating action with the appropriate database.

The visual language provides a very general and easily customizable interface to the underlying knowledge and data structures, and processes operating on them such as specific applications and general agents, including communications with, and remote procedure calls to, other systems. The language may be used as a 'wrapper' to existing applications available only in binary form, and it may also be used as an embedded component of other applications available in source form. A computer-supported cooperative work approach has been adopted from the outset so that maps may be shared across local and wide area networks and used for distributed project coordination.

5. INTEGRATION WITH OTHER SYSTEMS

The primary objective of the Mediator development is to provide a 'shell' technology for coordinating collaborative manufacturing projects across networks. However, the open architecture, modular, networked, distributed client, distributed server technology underlying the Mediator implementation is well-suited to vertical application with applications as shown on the right of Figure 1. The development of 'Mediator-aware' applications is attractive not only in increasing the functionality of Mediator, but also in allowing rapid prototyping of new applications. The effort required to develop good quality user interfaces is a major component of resource usage in most application development, and the use of the generic Mediator interfaces and protocols can increase the speed, and decrease the cost, of specialist system development.

One reason a generic visual language was chosen as a general-purpose interface is because graphical representations are used in many branches of science and engineering, and the Mediator interface is to be able to emulate many such representations through a single software system. For example, a research group focusing on optimal scheduling and rapid shop-floor reconfiguration could develop algorithms that interface to Mediator for purposes of user interaction, and run on a high-performance server anywhere on the network.

The existing Mediator graphic representation facilities could be used to provide an iconic interface giving a geometric, or more abstract, representation of the configuration and material flows. However, the primary project resources could be applied to the development of improved scheduling and reconfiguration algorithms and not diverted to user interface aspects of the system which, while very important, are not the primary focus of the research. For example, Figure 3 shows a bond graph from HUT's Δ functional design system [14] represented in the visual language, and Figure 4 shows a partial behavioral model from University of Tokyo's SYSFUND functional design system [15] similarly represented. The generic Mediator data structures can be imported from, and exported to, the specific systems through simple converters.

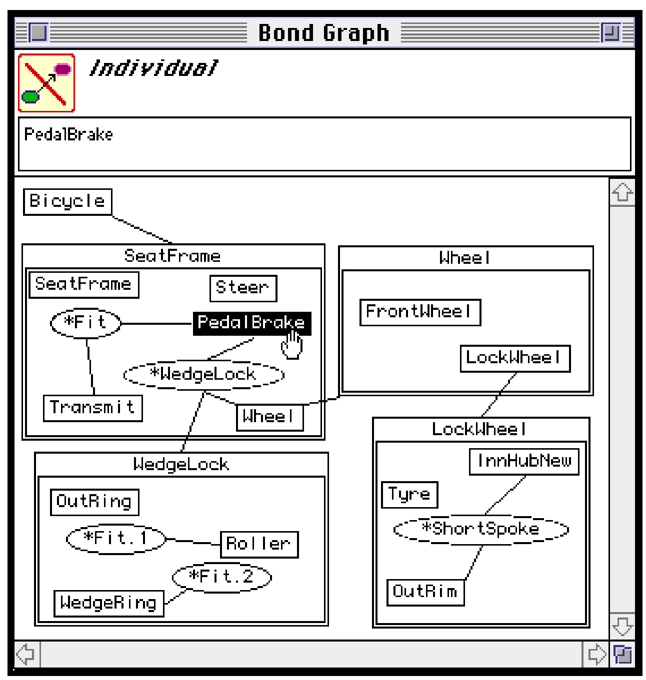

Figure 3 A Δ bond graph in Mediator

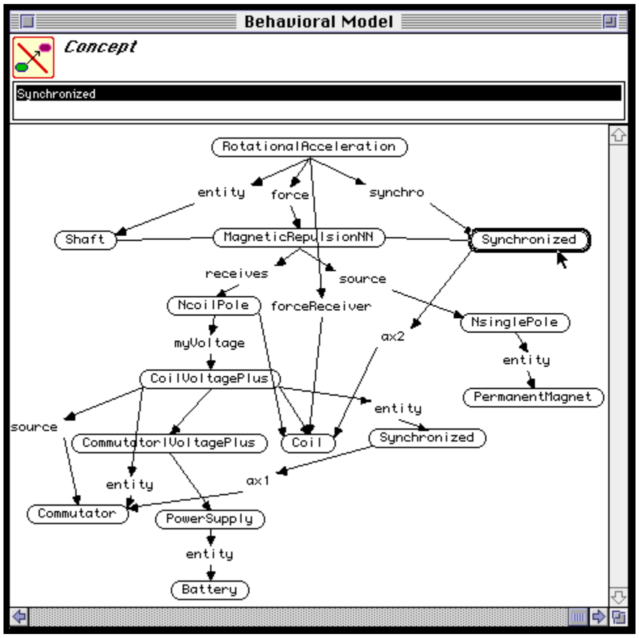

Figure 4 A SYSFUND behavioral model in Mediator

A semantic network visual language is already available for Mediator [6] that compiles into a KL-ONE-like knowledge representation system [5, 8] used to represent formal knowledge structures [9]. This has been used to operationalize a corporation's procedures manuals [18] in a way that is applicable, for example, to the regulatory aspects of manufacturing

6. MEDIATOR NETWORK PROTOCOLS

Figure 5 shows the way in which Mediator operates over a network. A server agent at a site manages a knowledge base consisting of a set of files from different applications. Concept maps are used to represent the files and relations between them. Files may be opened from the maps in the appropriate applications. Since the maps and hypermedia documents of Mediator are also files, the system can be used to support large-scale linked knowledge structures. Client agents at remote sites connect to server agents across the network and allow files to be accessed remotely in the same way as they are locally.

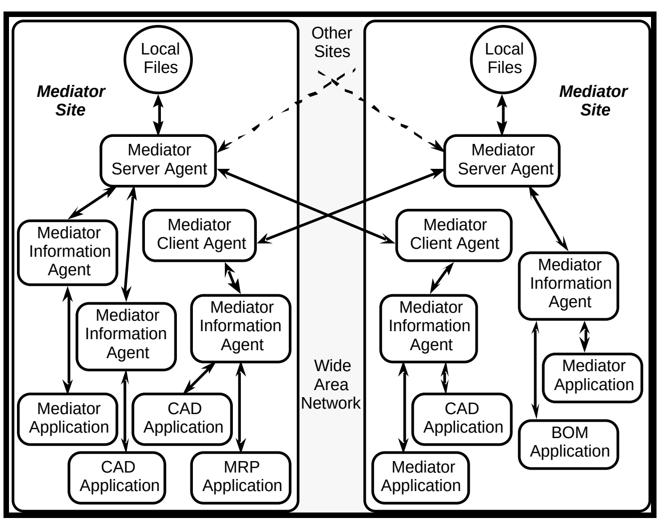

Figure 5 Mediator operation over a network

Figure 6 left shows a screen dump of a Mediator server agent managing files for three projects at one location. Figure 6 right shows a Mediator client agent at a remote location having linked to the server agent. The cursor in the right screen dump is over a node in the concept map and has changed to a button shape to indicate that the associated dataset will be opened if the mouse button is clicked.

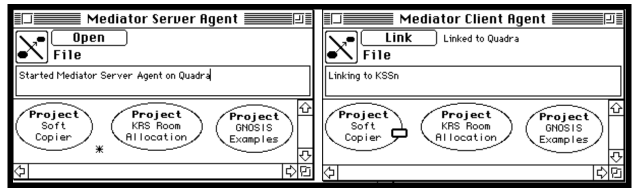

Figure 6 Client agent linked to server agent across a network

Figure 7 shows the dataset opened when the button is clicked. It is another concept map, but this time embedded in a document, giving the project structure from requirements through design to operational models, including a video demonstrating the prototype system in operation. The embedded concept map remains active. If the user selects this video through the "Open File" option in the popup menu as shown, or through the button cursor as shown above, then the video will open and play in a QuickTime viewer. The popup menu also allows the user to check on the application program appropriate to the file since it is being fetched from a remote site, and the local site may not have the appropriate application available. Embedding concept maps in active, printable documents [11] allows reports, manuals, etc, to be easily generated.

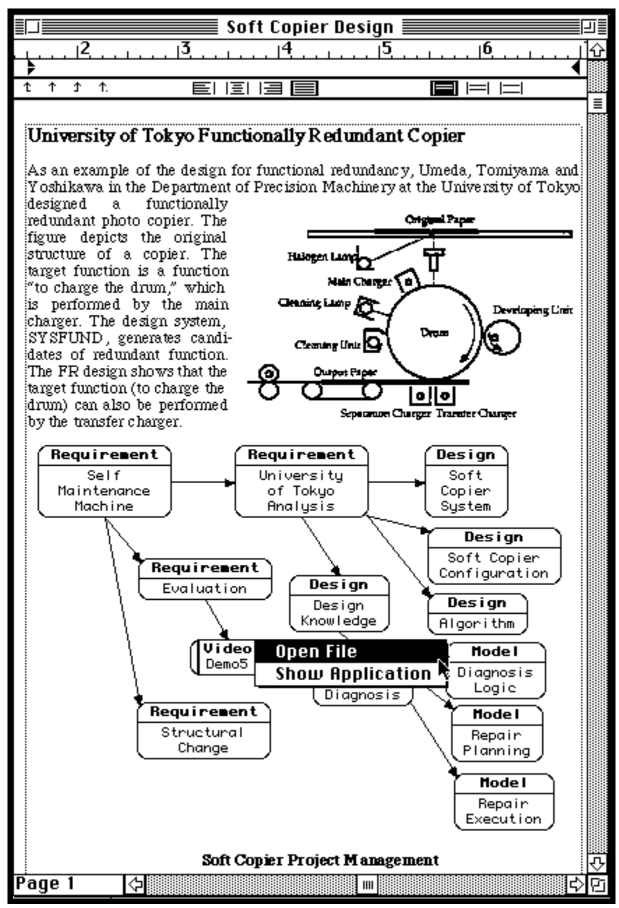

Figure 7 Project management concept map embedded in a document

7. IMPLEMENTATION ON WORLD-WIDE WEB

The initial Mediator prototype developed during the GNOSIS test case was implemented on the Apple Macintosh platform and operated over local area networks. It was issued on the CD-ROM that contained the final reports of the GNOSIS test case [13], and it was used to index these reports and the accompanying demonstration software and digital videos through layered concept maps accessing the heterogeneous collection of files that comprised the final reports, demonstrations and data sets . Figure 8 of the accompanying paper on GNOSIS shows Mediator in use to access material on the CD-ROM [10].

Since the completion of the GNOSIS test case in March 1994 we have had a systematic program of research designed to make Mediator technology widely available on a cross-platform basis operating through the Internet. The hypertext transport protocol (HTTP) [2] of the World-Wide Web (WWW) [3] was chosen for the low-level communications layer since HTTP servers are widely available for a range of platforms, as were associated browsers such as NCSA Mosaic and Netscape. In particular, the browsers support not only the hypertext markup language (HTML) [1] but also arbitrary data types through interface to other ‘helper’ applications such as our concept mapping tools. This makes it possible to implement the Mediator architecture of Figure 1 and the protocols of Figure 5 using standard protocols and application packages in large part. This is particularly significant because one objective of the research program in the past 18 months has been to move Mediator out of the research laboratory and make it widely available on the Internet to support the long-term IMS research program.

The main programming effort has been to port the concept mapping tools cross-platform so that they can act as client helpers to WWW browsers on any platform. Figure 8 shows Xcm operating under Motif and X-Windows under SunOS as a helper to Netscape.

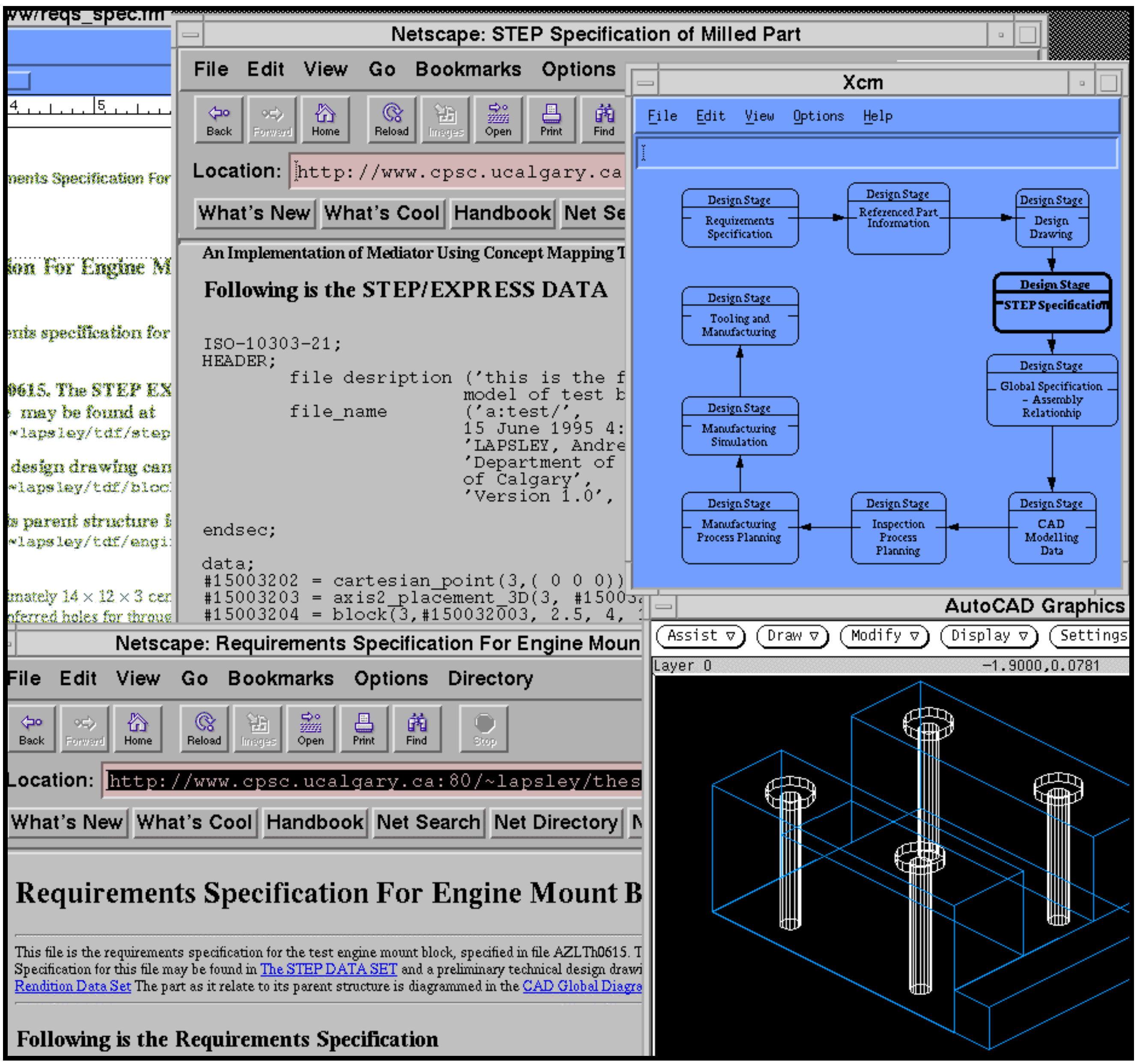

Figure 8 Screen dump of Mediator running on a UNIX workstation

At the top right of Figure 8 a Mediator concept map of a sequence of design processes for an engine mount has been opened. This was fetched by Netscape from an HTTP server on the Internet , and because the file extension was '.cm' the server and Netscape recognized it as a concept map data type and opened it in the Xcm application. A popup dialog associated with each node allows files to be specified that are associated with the nodes following the Mediator protocols described in Section 6. However, the file accesses are now specified through WWW uniform resource locators (URLs) which enables them to be fetched from any HTTP server on the Internet. At the top left the user has fetched a FrameMaker document containing the requirements specification for the engine mount which Netscape has opened in FrameMaker. At the lower left he has fetched an annotated requirements specification in HTML which has opened in Netscape and itself contains hypertext links to other files. At the top center the user has opened an HTML document containing the STEP specification of the mount, and at the lower right the AutoCAD drawing which Netscape has opened in AutoCAD. A similar screen would be generated on other platforms by the same sequence of actions.

8. CONCLUSIONS

Mediator, an open architecture information and knowledge management system has been described which provides a flexible technology to support the management of complex manufacturing environments. Its implementation on the World-Wide Web provides a powerful general tool for the support of the virtual manufacturing enterprise.

ACKNOWLEDGMENTS

This work was funded by Industry, Science and Technology, Canada, as part of the international IMS research program, and by the Natural Sciences and Engineering Research Council of Canada. We are grateful to many colleagues in IMS for their contributions to the Mediator project, in particular, Aldo Dagnino, Peihua Gu, Martti Mäntylä, Tetsuo Tomiyama and Moriki Toyama.

REFERENCES

[1] J. Barry, "The hypertext markup language (HTML) and the World-Wide Web: Raising ASCII text to a new level of usability," *Public-Access Computer Systems Review*, vol. 5, no. 5 pp. 5-62, 1994.

[2] T. Berners-Lee, Hypertext Transfer Protocol. CERN, Geneva. 1993.

[3] T. Berners-Lee, R. Cailliau, A. Luotonen, H.F. Nielsen, and A. Secret, "The World-Wide Web," *Communications ACM*, vol. 37, no. 8 pp. 76-82, 1994.

[4] T. Finin, J. Weber, G. Wiederhold, M. Genesereth, R. Fritzson, D. McKay, J. McGuire, S. Shapiro, and C. Beck, Specification of the KQML Agent-Communication Language. The DARPA Knowledge Sharing Initiative External Interfaces Working Group. 1992.

[5] B.R. Gaines, "Empirical investigations of knowledge representation servers: Design issues and applications experience with KRS," *ACM SIGART Bulletin*, vol. 2, no. 3 pp. 45-56, 1991.

[6] B.R. Gaines, "An interactive visual language for term subsumption visual languages," in *IJCAI'91: Proceedings of the Twelfth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann: San Mateo, California. p. 817-823, 1991.

[7] B.R. Gaines, "Manufacturing in the knowledge economy," in *Proceedings of ICOOMS'92: International Conference on Object-Oriented Manufacturing*. University of Calgary: Calgary. p. 19-36, 1992.

[8] B.R. Gaines, "A class library implementation of a principled open architecture knowledge representation server with plug-in data types," in *IJCAI'93: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann: San Mateo, California. p. 504-509, 1993.

[9] B.R. Gaines, "Experience with a class library for organizational modeling and problem solving," *Integrated Computer-Aided Engineering*, vol. 1, no. 2 pp. 93-107, 1993.

[10] B.R. Gaines and D.H. Norrie, "Knowledge systematization in the international IMS research program," in *Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics*. IEEE: New York. p. 958-963, 1995.

[11] B.R. Gaines and M.L.G. Shaw, "Open architecture multimedia documents," in *Proceedings of ACM Multimedia 93*. p. 137-146, 1993.

[12] B.R. Gaines and M.L.G. Shaw, "Concept maps indexing multimedia knowledge bases," in *AAAI-94 Workshop: Indexing and Reuse in Multimedia Systems*. AAAI: Menlo Park, California. p. 36-45, 1994.

[13] GNOSIS, GNOSIS: Intelligent Manufacturing Systems: IMS Test Case 7: Hybrid-CD, Macintosh (native), PC, Unix (ISO 9660). Knowledge Science Institute and Division of Manufacturing Engineering, University of Calgary, Canada. 1994.

[14] J.-K. Gui and M. Mäntylä, Assembly modeling on the basis of a mechanical design prototype. Laboratory for Information Processing Science, Helsinki University of Technology. 1993.

[15] M. Ishii, T. Tomiyama, and H. Yoshikawa, A synthetic reasoning method for conceptual design. Department of Precision Mechanical Engineering, The University of Tokyo. 1993.

[16] D. Karnopp, R.C. Rosenberg, and J.J. van Dixhorn, "Bond Graph Techniques for Dynamic Systems in Engineering and Biology," *Journal Franklin Institute*, vol. 308, no. 3, 1989.

[17] R. Kremer and B.R. Gaines, "Groupware concept mapping techniques," in *Proceedings SIGDOC'94: ACM 12th Annual International Conference on Systems Documentation*. ACM: New York. p. 156-165, 1994.

[18] R.C. Kremer, "Experience in applying KRS to an actual business problem," in *Proceedings of the Sixth AAAI Knowledge Acquisition for Knowledge-Based Systems Workshop*, J.H.G. Boose, B.R., Editor. University of Calgary: Calgary, Canada. p. 11-1-11-12, 1991.

[19] A.D. Kwok and D.H. Norrie, "Integrating multiple reasoning in intelligent agent systems," *Integrated Computer-Aided Engineering*, vol. 1, no. 2 pp. 83-90, 1993.

[20] A.D. Kwok and D.H. Norrie, "Intelligent agent systems for manufacturing applications," *Journal of Intelligent Manufacturing*, vol. , pp. 285-293, 1993.

[21] F.P. Maturana and D.H. Norrie, "A generic mediator for multi-agent coordination in a distributed manufacturing system," in *Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics*. IEEE: New York. p. 952-957, 1995.

[22] W. Reisig, *Petri Nets: An Introduction*, Berlin: Springer. 1985.

[23] J.F. Sowa, *Conceptual Structures: Information Processing in Mind and Machine*, Reading, Massachusetts: Adison-Wesley. 1984.