

# Supporting the Information Mesh

Karen R. Sollins

MIT Laboratory for Computer Science

Cambridge, MA 02139

sollins@lcs.mit.edu

As the networked world expands, it holds the promise of a new vision of human and computer connectivity. We envision a new model for the role of the network in a distributed computing environment, as the basis or core of the *Information Mesh*. Our early model of the network was a reliable stream of bits, facilitating such network activities as file transfer, remote login, and electronic mail. As networks became more prevalent, we incorporated them into our models of programming, as in the various forms of remote procedure call. In both of these cases, the major impact of networking on the operating system was real time constraints on the operating system; not only were there real time constraints based on application level tasks, but also the network itself presented a further real time constraint on the operating system.

As high-speed workstations and networks become more prevalent, we find ourselves able to provide a new model of the network, that will necessarily require abstractions in the operating system. This new model is one based on information that is long-lived and widely distributed; we envision upwards of  $10^{14}$  objects or nodes with lifetimes of up to 100 years.. Furthermore, the model incorporates the concept of relating or *linking* nodes of information to form the *information mesh*. It is this notion of links that must be understood at the operating system level, in order to participate in this large, interconnected mesh of information.

The model assumes that interesting information can be distributed around the world, and survives outside any particular application, application toolkit, or programming language runtime system. In addition, for the information to be interesting, it will be extremely long-lived. It may outlive ownership, storage media, file systems, and other services for locating the information, each to be replaced by successors. Thus, the *names* that occur in links must outlive ownership, storage facilities, and underlying organizations, yet be generally resolvable at any time.<sup>1</sup> Thus the name for a node at the other end of a link must contain no semantics that will restrict the size, mobility, or longevity of the node.

There are three other noteworthy aspects of links in need of system support: evolvable

---

<sup>1</sup>In this research, we take the position that it may not always be possible to resolve the name in a link from any point, even if the node still exists. This may be due to lack of network connectivity or inability to find a location service that knows about the node.

typing, ranges, and hints. First, the Information Mesh takes the position that links are typed. If it is important to express a relationship between two nodes of information, it is also important to be able to express something about the nature of that relationship. Thus, the fact that paper A *refutes* paper B should be distinguishable from the fact that paper A *is derived from* paper C. In addition, it may be that as paper A is being written, the author originally knew that it was related somehow to paper C, but not exactly how. As the paper evolves, the nature of the relationship becomes clear to the author. The author may decide on a particular meaning of derivative work, implying conceptual derivation. Previously, if the only definition of *derived-from* included the idea that the writing was derived from a previous piece of text and that the ideas expressed were the same, then the author would have a new private definition of the relationship. The new definition might later become widely accepted and defined as a global type, perhaps calling it *concept-derived-from*.

Ranges for links are a simpler concept and found frequently in hypertext systems. A link is defined as connecting a range within the nodes at each end. Thus, locally it is associated with some part of the node, and it points to some range of the remote node, definable in exported dimensions of the type of the remote node.

Hints are the system supported part of links that facilitate the actual location of a remote node. Hints may consist of such information as the source from which the name came (because it may know how to find the node), trusted locations servers for this sort of node, current ownership information, or even current storage repository. Hints are exactly that; they are not necessarily reliable. They may become less valuable with time and mobility of either end of a link. In order to provide flexibility and utility, hints must be mutable. It is the hints that may help to find a location service that in turn can locate the node.

Within the Information Mesh, computations or activities will take place. They will probably take one of three forms, all of which may be provided by system libraries and facilities in programming languages. First, the simplest form of mesh operation is one in which the focus of attention is changed, but the application remains the same. For example, if one is reading a document and wishes to follow a reference link, one might simply find oneself reading another document. Second, as the focus changes, the application itself might also change. In this case, if the reference were to an interactive video stream, as the focus changed, the application would also change. The third sort of operation is less specific, more global. Searching is a good example of it; although we envision other global operations as well, and intend to generalize this concept. In cases of such global operations, one may need to limit the scope in terms of the mesh. These are operations over regions of the mesh.

In order to maintain the integrity of a node of information, its links must be expressed in a separate link shadow, one shadow tightly coupled to each information node and recognizable by the operating system (although not necessarily the kernel). There are two reasons for the link shadow to be distinct from the node itself. The first is to maintain compatibility with existing applications. If the representation of the information were to change, many existing applications would become useless or require modification. Thus, we would be reluctant to include our most important and useful information in the mesh. Second, in order to facilitate location of a node, a link will contain hints to be used by a location service. With time these hints may change in order to remain useful. Separating, at least conceptually, the mutable part of links from a possibly immutable node of information is important.<sup>2</sup>

It will become important to include links and a native form of link shadow in each operating system, if the workstation running that operating system is to allow the user to collaborate in the Information Mesh. Because such information is generally much more useful if it is application and programming language independent, the nodes and links will exist and be managed by the system, or, in general, by the collection of workstations and servers.

Although the focus here is on link shadows, there are other parts of the Information Mesh infrastructure with which the operating system will need to interact. Some of these are (1) a model of typing of both nodes of information and links, (2) flexible and changing location services, (3) a model of security and access control, (4) the model of computation, and (5) a toolkit for organizing and viewing the organization of the mesh or regions of it.

This work is a synthesis and extension of ideas from a number of related areas of work, but does not fit into any one area. The related areas that will be mentioned here are distributed hypertext or hypermedia systems, large information systems, multidatabases, and knowledge representation systems. Only an incomplete sampling in these areas will be presented to provide the reader with an understanding of the distinctions between this work and any one of these areas. We will only allude to those works; the reader is referred to the references and related work for details.

The Information Mesh bears the most resemblance and perhaps could well be described as a global hypermedia system, with enhancements or extensions beyond the current state

---

<sup>2</sup>Related to this is that the number of links associated with a node may change with time. For example, if a journal which indexes articles, maintains backwards pointers from articles to the articles that reference them, the number of links in an article will generally increase over time. By keeping these in a shadow, managing them may be simplified.

of the art. The three such systems that bear the most resemblance are the IRIS Intermedia Project[5], Project Xanadu[9, 10], and the World Wide Web[2]. Intermedia runs in a distributed environment; it supports only links that imply inclusion of one object in another, and does not support wide distribution or a large environment. All link information is maintained in a single database, so that any modifications that might affect any links, will be reflected in those links immediately. It therefore would not scale to the size of the Information Mesh.

Project Xanadu addresses these issues and many more, but there are several significant differences between it and the Information Mesh. First, Xanadu and all its protocols are proprietary and therefore generally unavailable. The major implication of this is that one must use the Xanadu file system, type system, and all other aspects of Xanadu, exclusively. It does not support access to information that has not been created within the Xanadu world. Second, it is our goal to allow link types to evolve with time; Xanadu does not support this. Third, the names (addresses) in Xanadu have semantics imposed on their components. For example, since an address includes a logical address for the birthsite, although the binding of this address to a physical server might mutate, it cannot effectively be partitioned by migrating each of the objects created at a particular server to separate alternative servers.

Finally, the World-Wide Web supports the global nature for which we are aiming in the Information Mesh, but there are several other ways in which it is at variance with our intentions. At present links are untyped. In addition, the names contain some form of information about the “publisher” of an object. This is yet another form of imposition of semantics on a name, which may change with time, and therefore cause problems with finding and using extremely long-lived objects. It is probably the case that the Information Mesh has the most in common with the World-Wide Web.

Large information systems appear to be one of the new areas for great activity; these systems are springing up all over the place as system managers of large sites attempt to get a handle on large amounts of heterogeneous information, in support of their user communities. The World-Wide Web is one of these. In addition, two other distinctive ones are WAIS (the Wide Area Information Servers) from Thinking Machines[6, 7] and *archie*[4]. These two can be distinguished from each and represent two types of such services because WAIS encapsulates the files or documents it is indexing in WAIS servers, while *archie* is providing pointers to files, in *archie*'s case files available by anonymous FTP. In both cases, there is no linking. In other words, there is no system support for finding and following names within the files, as opposed to those in the indices of these two systems. On the

other hand, both are flexible about supporting a heterogeneous environment. The WAIS protocols are public, and *archie* utilizes anonymous FTP protocols for accessing any file system that supports it.

Databases is another area in which there is a great deal of overlap with the Information Mesh, especially distributed or federated databases. There are currently two major thrusts in the models for databases. Object-oriented databases as described by Atkinson et al.[1] bear the closer relationship to the Information Mesh and hypertext systems. Objects are typed and links between them are supported. This approach is new and little work has been done in creating multidatabases of object-oriented databases. Instead, as reported by Breibart[3], the work to date on creating unified database environments from multiple different environments has been in the area of relational databases. Here two approaches are taken: *global* or *federated*. In the first case, a global schema is developed, with each participating database utilizing a subschema of that global schema. In this case there is tight coupling and references between databases is possible. But there is little room for heterogeneity. The federated approach depends on each participating database to have define its external interfaces and which data is accessible, but because of the looser relationships, there is no effective way to link information across those boundaries. Thus neither approach is quite what is planned for the Information Mesh, but again there are similarities, as well as these differences.

Lastly, we consider the *Cyc* project[8] as representative a widely distributed system from the field of Knowledge Representation. Although Lenat and Guha envision a widely distributed system of knowledge or information, in which the knowledge and the relationships among components grow and change with the acquisition of more knowledge, there is an underlying philosophical difference between this approach and that of the Information Mesh, and the other large information systems, especially. The architects of knowledge-based systems have a clear idea of the nature of knowledge and are architecting a system in which they and other like-minded knowledge engineers define the nature of the information. The system is designed to take over and expand its horizons. This view is distinctly different from our view that we cannot possibly predict the nature or acquisition of the information that should be potentially accessible. In addition, more mechanistically, as with many of the above mentioned systems, *Cyc* takes as a given that the knowledge will be stored in *Cyc*, using *Cyc* mechanisms. In order to incorporate information, it must be in *Cyc*. Interoperability and heterogeneity are not part of the focus of *Cyc*. But what knowledge-based systems such as *Cyc* bring to the Information Mesh beyond distributed information is the idea of evolution of the knowledge, as represented in the links of the Information Mesh.

Thus, as with the other systems, and categories of information systems, the Information Mesh takes something from each, but is none of the above.

At present we are working on understanding the nature of links and the names in them. It is our intention to build a location service for finding nodes with the names in links and the link and link shadow abstractions, in order to begin building applications that can compute in the Information Mesh. We plan to work both from the approach of building the tools in order to build applications, and to build applications in order to learn about the facilities required to realize the Information Mesh. This research is in its earliest stages.

## References

- [1] M. Atkinson, F. Bancilhon, D. DeWitt, K. Dittrich, D. Maier, and S. Zdonik. The Object-Oriented Database System Manifesto. In *Proceedings of the 1989 Deductive Object-oriented Database Conference*, December 1989.
- [2] T. Berners-Lee, R. Calliau, J-F Groff, and Pollermann B. World-Wide Web: The Information Universe. *Electronic Networking*, 2(1):52–58, Spring 1992.
- [3] Y. Breibart. Multidatabase Interoperability. *SIGMOD Record*, 19(3), September 1990.
- [4] A. Emtage and P. Deutch. Looking towards archie 3.x. Distributed at the Internet Engineering Task Force Meeting, San Diego, March 1992.
- [5] B.J. Haan, P. Kahn, V.A. Riley, J.H Coombs, and N.K Meyrowitz. Iris Intermedia Services. *CACM*, 35(1):36–51, January 1992.
- [6] B. Kahle. Wide Area Information Servers Concepts. TMC 202, Thinking Machines Corp., Cambridge, MA, November 1989.
- [7] B. Kahle. An Information System for Corporate Users: Wide Area Information Servers. TMC 199, Thinking Machines Corp., Cambridge, MA, April 1991. Also in *ONLINE Magazing*, Aug. 1991.
- [8] D.B. Lenat and R.V. Guha. *Building Large Knowledge-Based Systems, Representation and Inference in the the Cyc Project*. Addison-Wesley Publishing Co, Inc., Reading, Massachusetts, 1989.
- [9] T.H. Nelson. *Literary Machines*. Microsoft Press, 87.1 edition, 1987.
- [10] T.H. Nelson. Managing Immense Storage. *Byte*, pages 225–238, January 1988.