B-NODES: A NEW METHOD FOR MODELLING DIGITAL LIBRARIES

S P Maj, D Veal, R White
Department of Computer Science, Edith Cowan University
Perth, Western Australia
p.maj@cowan.edu.au

Abstract

In order for digital libraries to provide a prompt and possibly worldwide service they must be based on a complex infrastructure of hardware and software. Whilst it is recognized that librarians may not be responsible for the design and implementation of this infrastructure they are often in managerial positions. In this capacity librarians must therefore not only be fully conversant with the various technologies but also be able to manage them in order to provide an appropriate quality of service. A wide variety of modeling techniques and associated metrics exist. However this wide range is in itself problematic and arguably adds to the complexity. Accordingly a new modeling method called Bandwidth-Nodes (B-Nodes) has been developed and tested. B-Nodes are simple, easy to use, diagrammatic and self-documenting. B-Nodes models are scalable and can be used to describe small systems or a global information infrastructure. They provide recursive decomposition and hence can be used to produce both simple models and complex models thereby controlling complexity. B-Nodes use abstraction and are therefore independent of underlying technologies. They are therefore applicable to old and new technologies and are likely to be valid for all future technological developments. It is possible to model a digital library using B-Nodes and use this technique to obtain key library management data for performance and capacity planning. Results to date strongly suggest this technique may be of value for the management of digital libraries.

Keywords

B-Nodes, modeling, digital libraries

1. Introduction

Whilst there is no single, agreed definition of the term ‘digital library’ the common theme is the establishment of networked, on-line information providing access to resources on either a local or global basis. Developments in computer and network technology have resulted in the demand for on-line services from digital libraries. Furthermore it is noted that progressively user-driven expectation is for the integration of digital libraries into the daily workflow of users.

> ‘*Recently, many bibliographic systems, medical databases, knowledge-based systems, and online books have been built to improve access to information and to support clinical decision making.*’ [22]

In this context access must therefore be further qualified in terms of Quality of Service (QOS). QOS is characterized by three main themes: security, availability and performance. At one extreme a digital library may simply provide access to information on a ‘best attempt’ basis. Clearly this level of QOS is inadequate for digital libraries that must support ‘real time’ usage during, for example, patient consultation in which medical decisions must be made. Furthermore, medical applications such as this require access to a wide range of resources. The Stanford Health Information Network for Education (SHINE) integrates a wide range of digital information (journals, books, bibliographic systems, medical images, video etc) in order to support both on-line medical decision-making and learning. Typically QOS is defined according to a predefined Service Level Agreement (SLA). SLA’s defined the upper and lower bounds of performance, availability and security. These three elements are fundamentally bound to the design and management of the complex infrastructure of hardware and software upon which digital libraries are based. However according to Munson,

> ‘*The paradox is that research on digital information systems has had little to say about efficiency*.’

And that,

> ‘*Thus, most research has focused on improving accuracy and functionality rather than performance*.’ [18]

Furthermore, effective evaluation techniques are yet to be applied in any coherent way to digital libraries, although a number of researchers are exploring the possibilities in this area [21]. Whilst it is recognized that librarians may not be responsible for the design and implementation of this infrastructure they are often in managerial positions. In this capacity librarians must therefore not only be fully conversant with the various technologies but also be able to manage them in order to provide an appropriate quality of service. In order to control this complexity both methods and models are used. However there exist a wide range of methods, which in itself is potentially problematic and arguably adds to the complexity of designing a digital library.

2. Methods and Modeling

A method is a collection of procedures, techniques, tools and documentation aids that provide guidance and assistance to system developers. A method consists of phases or stages that in themselves may consist of sub-phases. There exist a wide range of methods that include ad hoc [9], waterfall [20], participative [17], soft systems [4], prototyping [19], incremental [7], spiral [3], reuse [14], formal [2], rapid application development [13], object oriented [5] and software capability [8]. Regardless of the underlying theme of each information system method all methods must provide techniques for modeling data, processes, system functions. Some systems development methods only stress the technical aspects. It can be argued that this may lead to a less than ideal solution as these methods underestimate the importance and difficulties associated with the human element. Typically, for information systems the formalisms (mathematical framework and language) used to make models of systems are relatively weak. Often data-flow diagrams and entity-relationship diagrams are the only formalisms used. However, considerable emphasis is placed on interview techniques, database design etc. The Structure Systems Analysis and Design Method (SSADM) was evaluated as a method for developing an information system. SSADM is mandatory for UK central government software development projects.

This method is sponsored by the Central Computer and Telecommunications Agency (CCTA) and the National Computing Centre (NCC) thereby further ensuring its importance within the software industry within the UK. SSADM is a framework employing a wide range of techniques (Data Flow Diagrams, Entity Models, Entity Life Histories, Normalization, Process Outlines and Physical Design Control). SSADM is divided into six stages (Analysis, Specification of Requirements, Selection of System Option, Logical Data Design, Logical Process Design and Physical Design). The Physical Design translates the logical data design into the database specification and the logical process designs into code specifications. Whilst SSADM is recognized to be a highly structured method with numerous cross checks to ensure quality control there are problems. There are only a few tools to assist in performance management. Capacity planning is used to estimate the data storage requirements of the hard discs. It is possible to analyse the process specifications detailed in the analysis and calculate the processing load – typical units include Million Instructions Per Second (MIPS). However, no guidance is provided in the selection of hardware and the effect on performance. Furthermore it has been argued that many benchmark performance metrics are of questionable value [10]. Currently there appears to be no simple technique that will model the digital infrastructure (hardware and software) to determine if it will perform to an acceptable standard required by the analysis and design specifications. Accordingly modeling theory was examined.

3. Modeling Theory

Models are used as a means of communication and controlling detail. Diagrammatic models should have the qualities of being complete, clear and consistent. Consistency is ensured by the use of formal rules and clarity by the use of only a few abstract symbols. Leveling, in which complex systems can be progressively decomposed, provides completeness. According to Cooling [6], there are two main types of diagram: high level and low level. High-level diagrams are task oriented and show the overall system structure with its major sub-units. Such diagrams describe the overall function of the design and interactions between both the sub-systems and the environment. The main emphasis is ‘what does the system do’ and the resultant design is therefore task oriented. According to Cooling, ‘*Good high-level diagrams are simple and clear, bringing out the essential major features of a system’*. By contrast, low-level diagrams are solution oriented and must be able to handle considerable detail. The main emphasis is ‘how does the system work’. However, all models should have the following characteristics: diagrammatic, self-documenting, easy to use, control detail and allow hierarchical top down decomposition.

It is possible to represent computer and network technology as a series of increasing abstract levels [23]. At the lower level the technology consists of solid-state electronic switches. Such switches may be described with models directly relevant to engineers operating at this level of complexity. Switches are used to make logic gates (NAND, NOR etc) and can be modeled using symbolic Boolean algebra. The underlying switching technology is not relevant at this higher level of abstraction. Similarly logic gates can be used to implement combinatorial and sequential functional units such as Read Only Memory (ROM) circuits. At each level different models must be used. Furthermore there appears to be no simple modeling method for technological devices such as: PC, microprocessor, hard disc drive, network card, Local Area Network etc. Furthermore the use of benchmarking metrics is problematic.

4. B-Nodes

It is possible to model computer and network technology using Bandwidth-Nodes (B-Nodes) [12]. Using this model each B-Node (microprocessor, hard disc drive etc) can be treated as a data source/sink capable of, to various degrees, data storage, processing and transmission. Each B-Node can be treated as a quantifiable data source/sink with an associated, common, transfer characteristic (Mbytes/s). This approach allows the performance of every node and data path to be assessed by a simple, common measurement – bandwidth. Where Bandwidth = Clock Speed x Data Path Width with the common units of Frames/s (Mbytes/s). Even though technical detail is lost, this model is conceptually simple, controls detail by abstraction. The B-Node modeling method is a diagrammatic, abstract method that employs recursive decomposition. The abstraction allows the underlying complexity to be controlled whilst still communicating the essential features. The use of abstraction decouples the method from the technology. It is therefore valid not only for past but also current technologies. It is suggested it may well therefore be valid for future technological developments. The method is scalable and it is possible to model not only digital circuits, the components (microprocessor, electronic memory, hard disc drive etc), the computer itself and a network as B-Nodes [11]. Significantly the performance of each B-Node can simply be calculated and common units are used (Mbytes/s). It is therefore possible to evaluate the relative performance of heterogeneous devices using one simple modeling technique. The use of derived units allows other, more useful units to be derived. Advantages of this new modeling method include:

- Simple, easy to use, diagrammatic and self-documenting.

- Scaleable and can be used for small systems e.g. modules within a PC, a complete PC, a Local Area Network (LAN) or a global information system.

- Recursive decomposition allows complexity to be controlled.

- Fundamental performance metrics allow other, more meaningful, derived units to be used.

- Abstraction hence B-Nodes are independent of the underlying technology. It is therefore applicable to both old and new technologies and is likely to be valid for future technological developments.

B-Nodes have been used to model a digital library currently in use here as follows.

5. Digital Libraries

Digital libraries must be designed, managed and maintained to provide a specific quality of service. Standards and evaluation criteria are designed to ensure that the service offered meets the needs of '*the community to be served*' and that a '*series of related collections [is] developed ... and shaped over time for use not only by the present generation but by generations yet to come*' (Taylor, quoted in [15].

Service Level Agreements in conjunction with cost constraints and the technologies used are the primary determinants in performance and capacity planning. In this context a variety of models are used. The Business model defines the purpose of the organization; the Functional Model defines the navigational structures and the Customer Model is used to describe the user behavior patterns. The number of clients, type of resources requested, pattern of usage etc all determine the workload characteristics. Workload characteristics, in conjunction with the resource infrastructure model will determine site performance and whether or not the Service Level Agreements can be met.

Customer Behavior Modeling methods have been successfully used to determine aggregate metrics for E-Commerce web sites [16]. Whilst there are differences between E-Commerce and Digital Libraries it should be note that,

> ‘*Electronic Commerce (EC) and Digital Libraries (DL) are two increasingly important areas of computer and information sciences, with different user requirements but similar infrastructure requirements*.’ [1].

Certainly it is possible to obtain a wide variety of different performance metrics that include: Hits/s, Page Views/Day, Unique Visitors etc. However there are problems with using these metrics to define the characteristics of the required infrastructure. It is possible to translate Hits/Day to the performance required of a hard disc drive or the type of network hosting the web and application servers – but it is complex.

If each server is modeled as a B-Node then performance metric is bandwidth with units of Mbytes/s. If there are separate servers on a LAN it is possible to model the LAN as a B-Node using the same performance metric. The sub-modules of a server (microprocessor, hard disc, electronic memory etc) and also be modeled as B-Nodes, again using the same performance metric. Using a Customer Behavior Model Graph it is possible to characterize the navigational patterns of a group of users accessing the site i.e. the probabilities of each resource request (Entry, Home, Browse etc). Each resource request is implemented by means of a communication, which is in effect a collection of bytes. In effect, user performance metrics are converted by Mbytes/s. Given that common units are used, the performance of the entire infrastructure can be easily defined and evaluated. Conversely, it is possible to use units e.g. Hits/s that can be easily derived from the fundamental units of Mbytes/s. The performance of all the heterogeneous devices in the infrastructure can then be evaluated by a single, meaningful metric. From the number of users and their request pattern behavior it is possible therefore to quantify peak and surge capacities.

B-Nodes allow a simple performance model to be produced that is independent of the underlying technology. Certainly further work is needed to quantify the effects of compression and network operating systems. But the work to date strongly suggest the B-Node model is a useful tool for controlling the complexity of a digital library and also provide useful metrics for performance and capacity planning.

Conclusions

B-Nodes are a modeling technique that is easy to use, diagrammatic and self-documenting. They employ abstraction, recursive decomposition and scalability. B-Nodes can therefore be used to model a wide range of technologies (microprocessor, hard disc drive, LAN, PC, server etc). Using B-Nodes it was possible to model an on-line digital library based on standard client-server architecture. The use of B-Nodes appears to greatly simplify how the infrastructure is described and defined. The use of fundamental units (Mbytes/s) allows heterogeneous systems (microprocessor, server, LAN etc) to be easily evaluated. Other, more meaningful units such as Hits/s can be derived. B-Nodes are independent of the underlying technologies and may therefore be valid for future technological developments. Results to date strongly suggest this technique may be of value for the management of digital libraries.

References

[1] N. Adam and Y. Yesha, *Strategic Directions in Electronic Commerce and Digital Libraries: Towards a Digital Agora*, ACM Computing Surveys, 28 (1996), pp. 818-835.

[2] D. Andrews and D. Ince, *Practical Formal Methods with VDM*, McGraw Hill, New York, 1991.

[3] B. W. Boehm, *A software development environment for improving productivity*, Computer, 17 (1984), pp. 30-42.

[4] P. B. Checkland, *Systems Thinking, Systems Practice*, John Wiley, Chichester, 1981.

[5] P. Coad and E. Yourdon, *Object-oriented Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1991.

[6] J. E. Cooling, *Software Design for Real-Time Systems*, Chapman and Hall, Padstow, Cornwall, 1991.

[7] T. Gibb, *Principles of Software Engineering Management*, Wesley, Reading, MA, 1988.

[8] W. S. Humphrey, *Managing the Software process*, Addison-Wesley, Reading, MA, 1990.

[9] G. W. Jones, *Software Engineering*, Wiley, New York, 1990.

[10] S. P. Maj and D. Veal, *Architecture Abstraction as an aid to Computer Technology Education,, American Society for Engineering Education (ASEE) Annual Conference & Exposition*, St Louis, MO, 2000.

[11] S. P. Maj, D. Veal and A. Boyanich, *A New Abstraction Model for Engineering Students*, in Z. Pudlowski, J., ed., *4th UICEE Annual Conference on Engineering Education*, UNESCO International Centre for Engineering Education (UICEE), Faculty of Engineering, University of Melbourne, Bangkok, Thailand, 2001, pp. 200-203.

[12] S. P. Maj, D. Veal and P. Charlesworth, *Is Computer Technology Taught Upside Down?*, in T. J, ed., *5th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education*, ACM, Helskinki, Finland, 2000, pp. 140-143.

[13] J. Martin, *Rapid Application Development*, Macmillan, New York, 1991.

[14] Y. E. Matsumoto and Y. E. Ohno, *Japanese Perspectives in Software Engineering Practice*, Addison-Wesley, Reading, MA, 1989.

[15] P. McCauley, *From librarian to cybrarian: coping with accelerating changes in libraries*, Ohio Media Spectrum, 4 (2000), pp. 31-36.

[16] D. A. Menasce, V. A. F. Almeida, R. C. Fonseca and M. A. Mendes, *A Methodology for Workload Characterization for E-Commerce Servers,, 1999 ACM Conference in Electronic Commerce*, ACM, Denver, CO, 1999, pp. 119-128.

[17] E. Mumford and M. Wier, *Computer Systems in Work Design - the ETHICS Method*, Associated Business Press, London, 1979.

[18] E. V. Munson, *Performance Issues in Digital Information Systems: introduction to a special issue*, Journal of Digital Information, 1 (2000), pp. 1-2.

[19] J. D. Nauumann and A. M. Jenkins, *Prototyping: the new paradigm for systems development*,, *MIS Quarterly*, 1982.

[20] W. W. Royce, *Managing the development of large software systems: concepts and techniques*,, *WESCON*, 1970.

[21] T. Saracevic, *Digital library evaluation: Toward an Evolution of Concepts*, Library Trends, 49 (2000), pp. 350-369.

[22] M. C. Tsai and K. L. Melmon, *Digital Library for Education and Medical Decision Making*,, *3rd ACM Conference on Digital Libraries*, Pittsburgh, PA, USA, 1998.

[23] A. B. Tucker, B. H. Barnes, R. M. Aiken, K. Barker, K. B. Bruce, J. T. Cain, S. E. Conry, G. L. Engel, R. G. Epstein, D. K. Lidtke and M. C. Mulder, *A Summary of the ACM/IEEE-CS Joint Curriculum Task Force Report, Computing Curricula 1991*, Communications of the ACM, 34 (1991).