

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CISCO SYSTEMS, INC.
Petitioner

IPR2023-00537
Patent No. 7,916,630

DECLARATION OF INGRID HSIEH-YEE, PH.D.
UNDER 37 C.F.R. § 1.68 IN SUPPORT OF PETITION
FOR *INTER PARTES* REVIEW

TABLE OF CONTENTS

I.	Introduction	3
A.	Qualifications and Professional Experience	3
B.	Scope of This Declaration.....	11
C.	Evidence Considered in Forming My Opinions	11
II.	Authenticity and Public Availability of <i>Silva</i> (Ex.1005)	13
A.	Authentication.....	13
B.	Library of Congress Copy.....	14
C.	Library of Congress Records	16
D.	Cornell University Copy	20
E.	Cornell University Library Records.....	21
F.	Tulsa University Records.....	24
G.	Usage Record	27
H.	Summary of My Opinion on <i>Silva</i>	28
III.	Conclusion.....	28

I. Introduction

I, Ingrid Hsieh-Yee, Ph.D., do hereby declare as follows:

1. I have been retained as an independent expert witness on behalf of Cisco Systems, Inc. (“CISCO”) for proceedings before the United States Patent and Trademark Office (“PTO”) regarding the authenticity and public availability of certain prior art reference in relation to US Patent No. 7,916,630.

2. I am being compensated for my work in this matter at my customary hourly rate. I am also being reimbursed for any reasonable expenses associated with my work and testimony in this investigation. My compensation is not contingent on the results of my study, the substance of my opinions, or the outcome of this matter.

A. Qualifications and Professional Experience

3. My complete qualifications and professional experience are described in my academic *curriculum vitae* (**Appendix A**). The following is a brief summary of my relevant qualifications and professional experience.

4. I was a professor in the Department of Library and Information Science at the Catholic University of America for 32 years, retired in August 2022, and was named professor emerita in December 2022. I have experience working in an academic library, a medical library, and a legislative library. I hold a Ph.D. in Library and Information Studies from the University of Wisconsin-Madison and a

Masters in Library and Information Studies from the University of Wisconsin-Madison.

5. I am an expert on library cataloging and classification and have published two editions on this subject, *Organizing Audiovisual and Electronic Resources for Access: A Cataloging Guide* (2000 and 2006). I taught a variety of courses, including Cataloging and Classification, Advanced Cataloging and Classification, Organization of Internet Resources, Organization of Information, Digital Content Creation and Management, Internet Searches and Web Design, Information Literacy Instruction, Advanced Information Retrieval and Analysis Strategies, and The Information Professions in Society. I am familiar with metadata schema design and implementation. In my teaching I covered the design and implementation of metadata in databases, search engines, digital repositories, digital libraries, and digital archives. I also covered how the way we organize information resources affects the discovery and access to digital resources on the Internet. My research interests cover cataloging and classification, information organization, metadata, information retrieval, information architecture, digital collections, scholarly communication, user interaction with information systems, and others.

6. I am very familiar with a library cataloging encoding standard known as the “Machine-Readable Cataloging” standard, also known as “MARC,” which became the national standard for sharing bibliographic data in the United States by

1971 and the international standard by 1973. MARC is the primary communications protocol for the transfer and storage of bibliographic metadata in libraries. Experts in my field reasonably rely upon MARC records when forming their opinions.

7. A MARC record consists of several fields, each of which contains specific data about the work. Each field is identified by a standardized, unique, three-digit code corresponding to the type of data that follows. **Appendix B** is a true and correct copy of Parts VII to X of “Understanding MARC Bibliographic: Machine-Readable Cataloging” (<http://www.loc.gov/marc/umb/um07to10.html>) from the Library of Congress that explains commonly used MARC fields. *Understanding MARC Bibliographic* was originally published by the Follett Software Co. and copyrighted in 1988. Editions in 1989, 1990, 1994, 1998, 2000, 2003 and 2009 include changes. The latest edition was copyrighted in 2009. This document explains commonly-used MARC fields. For example, the personal author of the work is recorded in Field 100, the title is recorded in Field 245, publisher information is recorded in Field 260, the physical volume and characteristics of a publication are recorded in Field 300, and topical subjects are recorded in the 650 fields.

8. The Online Computer Library Center (OCLC) is the largest bibliographic network of the world, with more than 534 million records and thousands of member institutions (many of which are libraries of some type) in more

than 100 countries. OCLC was founded in 1967 to promote and support library cooperation. According to the “Third Article, Amended Articles of Incorporation of OCLC Online Computer Library Center, Inc.,” OCLC was created “to establish, maintain and operate a computerized library network and to promote the evolution of library use, of libraries themselves, and of librarianship, and to provide processes and products for the benefit of library users and libraries, including such objectives as increasing availability of library resources to individual library patrons and reducing the rate of rise of library per-unit costs, all for the fundamental public purpose of furthering ease of access to and use of the ever-expanding body of worldwide scientific, literary and educational knowledge and information” (source: <https://www.oclc.org/content/dam/oclc/membership/articles-of-incorporation.pdf>).

9. OCLC members can contribute original cataloging records in MARC to the system or derive cataloging records from existing records, an activity referred to as “copy cataloging.” When an OCLC participating institution acquires a work, it can create an original MARC record for the work in OCLC’s Connexion system (a system for catalogers to create and share MARC records), and the system will automatically generate a code for the date of record creation in the yymmdd format in Field 008, and the creating library’s OCLC symbol is recorded in subfield “a” of the 040 field. Once the MARC record is in Connexion, it becomes available to other OCLC members for adoption to their local online catalogs (*i.e.*, copy cataloging).

10. After a MARC record is created in Connexion, it also becomes searchable and viewable on WorldCat, which is a free web portal for users to explore the collections of more than 10,000 libraries worldwide. The record in WorldCat, however, is not presented in MARC fields. Instead, the data elements are labeled to help users interpret the record. Thus, the information stored in MARC records in Connexion is available to the interested public through the user-friendly and free web portal WorldCat.

11. Library online catalogs gained acceptance in the early 1980s and many libraries migrated their systems to the World Wide Web in the mid-1990s. Library online catalogs are based on MARC records that represent their collections in order to help the public understand what materials are publicly accessible in those libraries. Most libraries with online catalogs have made their catalogs freely available on the Web. These online catalogs offer user-friendly search interfaces. Strong user interest in keyword searches and the popularity of Google have led to the “googlization” of library search systems. As a result, many library catalogs now provide a single search box for users to conduct keyword searches, with additional support for searches by author, title, subject terms, and other data elements such as ISBN (International Standard Book Number). Library catalogs these days also offer features for users to narrow their search results by language, year, format, and other elements. Many libraries display MARC records on their online catalogs with labels

for the data elements to help the public interpret MARC records. Many libraries also offer the option to display MARC records in MARC fields.

12. Libraries create MARC records for works they acquire, including books, serials, motion pictures, and publications in other formats. MARC records are representations of materials held by libraries and are critical for the discovery and access to those materials. Monograph cataloging is fairly common in libraries, and most libraries make a newly cataloged monograph available to the public soon after the cataloging work is completed, usually within a week. Libraries can create original cataloging records or use an existing record in OCLC to create a copy cataloging record. As soon as the cataloging record is completed, it is added to the library's online catalog for users. If the record is an original record, it is also entered into OCLC WorldCat. If it is a copy cataloging record, the library's holding symbol is attached to the existing original record in OCLC WorldCat to facilitate searching and interlibrary loan.

13. The cataloging of serials and the serial check-in process are discussed here to show how libraries usually provide access to newly received serial issues. According to the glossary of the *RDA: Resource Description and Access* cataloging standard, a serial is "a mode of issuance of a manifestation issued in successive parts, usually bearing numbering, that has no predetermined conclusion. A serial includes a periodical, monographic series, newspaper, etc." Because the publisher of a serial

makes new issues of the serial available successively, a customary cataloging practice is to create one bibliographic record for the serial, and the serial record (encoded in MARC) typically provides information on the beginning date and frequency of the serial, not the dates of individual issues. In other words, libraries typically do not create MARC records for individual issues of a serial. Instead, they rely on a serial check-in system to track the receipt of new issues. A common check-in practice is to date stamp a new issue when it arrives. This practice has become automated since the late 1990s, and libraries now vary in how they share the receipt date of a new serial issue with the public. Some libraries use a date stamp, some affix a label to indicate the receipt date, some pencil in the receipt date, and some do not provide the information to the public.

14. The serial check-in process usually takes less than an hour, and one of the steps involves placing a date stamp on the new issue to document the date the issue is checked in. After that, the holdings information of the serial is updated in the library's catalog so that users know which issues are available for request or access. After serial check-in is completed, the new issue is placed on the shelf with the previous issues of the serial. Libraries with a public periodical room typically place new issues in the periodical room for easy user access. Because information presented in serials often reflects latest discovery, a general practice of libraries is to

make new issues of serials available for user access soon after they are checked in, usually within a week.

15. Conference proceedings can be cataloged as monographs or serials, depending on a library's policy. Libraries usually choose to catalog conference proceedings as monographs because records for monographs are easier for users to discover and understand, and access to individual conference proceedings becomes easier. Once a record for a conference proceedings is found, users can see the title, conference name, subjects, and related contributors clearly identified in the record and the call number will enable users to access the desired title quickly. Conference proceedings cataloged as monographs are treated as regular monographs, and the newly cataloged monographs typically become publicly accessible soon after the cataloging work is completed, usually within a week. When conferences have the same conference name over time and are held on a regular basis, libraries may choose to catalog the proceedings of such conferences as serials. This means that only one serial record is created for all the proceedings of a named conference, and no individual records are created for newly received conference proceedings. This approach saves cataloging time, and newly received conference proceedings are made available for user access soon after serial check-in, usually within one week.

16. I am personally familiar with many online catalogs, databases, and search engines. In preparing for this declaration, I used the following authoritative information systems to search for records:

- Google Scholar (<https://scholar.google.com>)
- Online catalog of the Library of Congress (<https://catalog.loc.gov/>)
- Online catalog of the library of Cornell University (<https://newcatalog.library.cornell.edu/>)
- Online catalog of the library of the University of Tulsa (<http://library.utulsa.edu/>)
- WorldCat (<https://www.worldcat.org>)

B. Scope of This Declaration

17. I have been asked to offer expert opinion on the authenticity and public availability date of the following document:

Silva, L. M., & Silva, J. G., (September 1993), “DIP: distributed diagnosis protocol,” *Microprocessing and Microprogramming*, vol. 38, nos. 1-5, pp. 171-178, **Ex.1005** (“Silva”).

C. Evidence Considered in Forming My Opinions

18. In the preparation of this declaration, I have reviewed the documents referenced below and any other documents I reference herein, and each of these is a

type of material experts in my field would reasonably rely upon when forming their opinions:

- (1) The document referenced above in Section I.B;
- (2) Copy of *Silva* obtained from the Library of Congress, **Appendix 1005A**;
- (3) Bibliographic and MARC records for *Microprocessing and Microprogramming* that contains *Silva*, available at <https://lcn.loc.gov/86641015> from the online catalog of the Library of Congress, accessed and obtained on November 14, 2022, **Appendix 1005B**;
- (4) Copy of *Silva*, obtained from the library of Cornell University on November 17, 2022, **Appendix 1005C**;
- (5) Bibliographic and MARC records for *Microprocessing and Microprogramming* that contains *Silva*, available at <https://newcatalog.library.cornell.edu/catalog/287221> from the online catalog of the library of Cornell University, accessed and obtained on November 17 and December 14, 2022, **Appendix 1005D**;
- (6) Bibliographic and MARC records for *Proceedings EUROMICRO*

93 *Open System Design Hardware, Software and Applications; Barcelona, September 6-9, 1993* that contains *Silva*, available at <http://library.utulsa.edu/record=b1005680> from the online catalog of the University of Tulsa, accessed and obtained on November 15, 2022, **Appendix 1005E**.

II. Authenticity and Public Availability of *Silva* (Ex.1005)

A. Authentication

19. **Ex.1005** is a true and correct copy of “DIP: distributed diagnosis protocol,” (“*Silva*”), by Silva et al., *Microprocessing and Microprogramming*, vol. 38, nos. 1-5 (September 1993), pp. 171-178, that I obtained from the publisher Elsevier. When I began my research on this article I searched Google Scholar by the title of *Silva* for records and found a link that led me to the publisher’s webpage for *Silva* at <https://www.sciencedirect.com/science/article/abs/pii/016560749390140G>. I followed the “Purchase PDF” link to purchase the article from Elsevier. This publisher copy is presented as **Ex.1005** in this declaration.

20. Page 1 of **Ex.1005** shows the title, authors, abstract and four keywords for *Silva*: Distributed Diagnosis, Fault Tolerance, Membership Protocols, and Network Monitoring. The top of the page shows this article appeared in “Microprocessing and Microprogramming 38 (1993) 171-177, North-Holland.” **Ex.1005** shows *Silva* has eight sections, seven figures and 29 references.

B. Library of Congress Copy

21. To determine the availability of *Silva* in libraries I searched WorldCat by “Microprocessing and microprogramming” for records, and a WorldCat record at <https://www.worldcat.org/title/7324063> shows the Library of Congress as one of the holding libraries. I then searched the online catalog of the Library of Congress for records, and the search result confirmed the holdings information. The Library of Congress records informed me that this journal had a call number of “QA76.5 .E89a” and users should request access to issues at the “Jefferson or Adams Building Reading Rooms.” I then obtained a scanned copy of *Silva* from the Library of Congress, with the assistance of Wisconsin TechSearch, a document delivery service based at the University of Wisconsin-Madison. This scanned copy is presented as **Appendix 1005A** in this declaration.

22. **Appendix 1005A** is a true and correct copy of *Silva* that I obtained from the Library of Congress. It includes the front matter of vol. 38, nos. 1-5 (September 1993) of *Microprocessing and Microprogramming* that contains *Silva* and the *Silva* article.

23. Page 1 of **Appendix 1005A** shows the covers and spine of a bound volume labelled as “Micro-Processing and Micro-Programming, 38-39, SEPT. – DEC. 1993, QA 76.5 .E89A.” Page 2 is the issue cover that shows the journal title and identifies this issue as “Volume 38, Numbers 1-5, September 1993” and shows

the journal's ISSN (International Standard Serial Number) is "0165-6074," its CODEN (six-character unique identifier of serial) is "MMICDT" and volume 38, nos. 1-5 contains pages "1-862." The cover explains that this journal is "The EUROMICRO Journal" and this volume contains "Proceedings EUROMICRO 93 Open System Design: Hardware, Software and Applications, Barcelona, September 6-9, 1993." The bottom of the issue cover shows the journal is "[p]ublished for EUROMICRO by NORTH-HOLLAND (a division of Elsevier Science Publishers B.V.)." Page 3 is a short title page. Page 4 is the copyright page that shows this volume has a 1993 copyright date and Elsevier Science Publisher B.V. is the copyright holder. Page 5 is the conference information page that carries a date stamp of "Library of Congress AUG 24 1993 Copyright Office." Page 5 also shows EUROMICRO 93 is the nineteenth EUROMICRO conference that took place in Barcelona from September 6 to 9, 1993, and the main theme of this conference is "Open System Design: Hardware, Software and Applications." Page 6 is the issue title page that shows the same title, issue number, and publisher information as the cover. Page 7 is the copyright page that shows the conference proceedings has a 1993 copyright date, and EUROMICRO Association for Microprocessing and Microprogramming is the copyright holder of the conference proceedings. It also shows this serial is published 15 times a year. Page 8 is the conference proceedings title page that shows the theme, conference information, editors, and publisher of the

proceedings. Pages 10 to 17 (internal pages xvi to xxiii) are the table of contents and page 10 (internal page xvi) shows *Silva* is included in B3 and begins on page 171. Pages 18 to 25 (internal pages 171 to 178) are the *Silva* article. I have closely compared the Library of Congress copy of *Silva* (**Appendix 1005A**) with the publisher's copy (**Ex.1005**) and found them to be the same.

24. **Appendix 1005A** is in a condition that creates no suspicion about its authenticity. Specifically, this copy is not missing any intermediate pages, the text on each page appears to flow seamlessly from one page to the next, and there are no visible alterations to the document. Furthermore, this copy of *Silva* was found in the custody of a national library, a place where, if authentic, it would likely be found. I therefore see no reason to question the authenticity of this copy of *Silva*.

C. Library of Congress Records

25. **Appendix 1005B** is a true and correct copy of the bibliographic record and MARC record for *Microprocessing and Microprogramming* whose vol. 38, numbers 1-5 (September 1993) contains *Silva*. I personally located, identified, and obtained these records on November 12, 2022, from the online catalog of the Library of Congress. This is the type of record experts in my field would reasonably rely upon when forming their opinion.

26. Pages 1 to 3 of **Appendix 1005B** show the bibliographic record and holdings information of *Microprocessing and Microprogramming* that contains

Silva. It shows the main title of this journal, North-Holland of Amsterdam as the publisher from 1981 to 1996, and the publication frequency from 1987 to 1993 as fifteen numbers a year. The record also shows “0165-6074” as the journal’s ISSN and “QA76.5 .E89a” as the call number.

27. Pages 4 to 6 of **Appendix 1005B** are the MARC record and holdings information. The first six digits of Field 008 show the MARC record was created on “810410” (*i.e.*, April 10, 1981). This date is followed by a “d” code that indicates the journal has ceased publication, and the next eight digits “19811996” indicate the journal began in 1981 and ceased publication in 1996. Subfield |a of Field 040 shows the record was originally created by “NDD” and subfield |d indicates “DLC” modified the original record to create a copy cataloging record. According to the *Directory of OCLC Members* (<https://www.oclc.org/en/contacts/libraries.html>), “NDD” is the OCLC library symbol for the Duke University Library, and “DLC” is the OCLC library symbol for the Library of Congress. Subfield |a of Field 042 shows that this record is verified by “pcc” (Program for Cooperative Cataloging of the Library of Congress), meaning name and series access points have been verified through the appropriate national level authority file.

28. Field 022 of the MARC record shows the ISSN of this journal is “0165-6074,” which is the same as the ISSN on the issue cover of **Appendix 1005A**. Field 030 shows “MMICDT” as the CODEN of this journal. Field 245 shows the journal

title, Field 260 shows North-Holland published this journal from 1981 to 1996, and Field 300 shows the journal has a total of 35 volumes. Field 362 shows the journal ran from “Vol. 7, no. (Jan. 1981)” to “v. 41, no. 10 (June 1996)” and Field 321 shows the publication frequency from 1987 to 1993 was fifteen numbers a year. Field 550 shows the journal was published for Euromicro, which is included in Field 710 of the MARC record to help users discover this journal. Fields 776 show this journal is also available online.

29. Subjects of this journal are represented in Field 050 by a Library of Congress Classification (LCC) number of “QA76.5,” which represents general works on digital computers (source: Library of Congress Classification - Schedule Q at https://www.loc.gov/aba/publications/FreeLCC/LCC_Q2020TEXT.pdf), and in Field 082 by a Dewey Decimal Classification (DDC) number of “004.16” that represents “personal computers” (source: <https://www.oclc.org/content/dam/oclc/webdewey/help/000.pdf>). Subjects are also represented by Library of Congress subject headings in two Field 650s, with “Microprocessors” and “Microprogramming” as main topics, each followed by “Periodicals” as a form subdivision (encoded in subfield “|v”) to indicate these topics are treated in a journal. These subject representations in the MARC record enable users to conduct subject searches to discover this journal.

30. This MARC record shows that *Microprocessor and Microprogramming* was a long-running journal, and the MARC record makes the journal discoverable at the online catalog of the Library of Congress. The Library of Congress copy of *Silva* (**Appendix 1005A**) shows a date stamp of “AUG 24 1993,” meaning this issue was received by the Library of Congress in August 1993, so the journal would have been discoverable in their online catalog by at least that time, if not earlier. Users would have been able to conduct subject searches by the LCC and DDC numbers and by terms in the Library of Congress subject headings to discover this journal. They would also have been able to search for the journal by title, ISSN, CODEN and Euromicro.

31. The date stamp on the Library of Congress copy of *Silva* (**Appendix 1005A**) has the appearance and distinctive characteristics of a typical check-in date stamp utilized by the library. As discussed earlier, libraries usually create one record for an entire journal, instead of creating records for individual issues of a journal, and they rely on a serial check-in system to monitor the receipt of new issues and update holdings information for users. Because journals publish latest research, the customary practice of libraries is to make a newly received issue available soon after serial check-in, usually within a week. At the Library of Congress this would mean that the newly checked-in issue would be placed in the Newspaper and Current Periodical Reading Room for easy user access soon after the serial check-in process

is completed. It is therefore my opinion that the issue containing *Silva* would have been publicly accessible by August 31, 1993, at the Library of Congress.

D. Cornell University Copy

32. As evidence that *Silva* was publicly accessible in other libraries, I searched WorldCat by the journal title for records and identified the library of Cornell University as another holding library. I searched their online catalog for records for the journal, and the search result confirmed the holdings information. I then obtained a copy of *Silva* from the library of Cornell University via Wisconsin TechSearch, a document delivery service based at the University of Wisconsin-Madison. This copy is presented as **Appendix 1005C** in this declaration.

33. **Appendix 1005C** is a true and correct copy of *Silva*. Page 1 is the issue cover, and page 2 is the conference information page that shows a date stamp of “Engineering Library AUG 25 1993 Cornell University.” Page 3 is the copyright page, page 4 is the title page of the conference proceedings, pages 5 to 13 are the table of contents, and pages 14 to 21 are the full text of *Silva*. I have closely compared the Cornell copy of *Silva* (**Appendix 1005C**) with the publisher’s copy (**Ex.1005**) and the Library of Congress copy (**Appendix 1005A**), and found the full text of *Silva* in these three copies is the same, and the front matter in the Cornell copy is the same as the pages included in the Library of Congress copy.

34. **Appendix 1005C** is in a condition that creates no suspicion about its authenticity. Specifically, this copy is not missing any intermediate pages, the text on each page appears to flow seamlessly from one page to the next, and there are no visible alterations to the document. Furthermore, this copy of *Silva* was found in the custody of an academic library, a place where, if authentic, it would likely be found. I therefore see no reason to question the authenticity of this copy of *Silva*.

E. Cornell University Library Records

35. **Appendix 1005D** is a true and correct copy of the bibliographic record and MARC record for *Microprocessors and Microprogramming* whose vol. 38, numbers 1-5 (September 1993) contains *Silva*. I personally located, identified, and obtained these records on November 17 and December 14, 2022, from the online catalog of the library of Cornell University. This is the type of record experts in my field would reasonably rely upon when forming their opinion.

36. Pages 1 and 2 of **Appendix 1005D** are the bibliographic record that shows *Microprocessing and Microprogramming* is a journal published for Euromicro by North-Holland, starting in 1981, and its ISSN is “0165-6074.” The “Availability” area shows the call number for this journal is QA76.5 .E89 and the library has volumes 7 to 41 published from 1981 to 1995/1996. The “Librarian View” link on page 2 leads to the MARC record, presented on page 3.

37. The first six digits of Field 008 of the MARC record (**Appendix 1005D**) show the record was created on “810730” (*i.e.*, July 30, 1981), and the “c19819999” code following the record creation date indicates this journal is a continuing resource that is ongoing. As discussed above, the Library of Congress MARC record indicates this journal ceased publication in 1996, but it seems that the Cornell MARC record was created in 1981 and has not been updated to reflect this change. Subfield “a” of Field 040 shows “NIC” created the original MARC record. According to the *MARC Code List for Organizations* (<https://www.loc.gov/marc/organizations/org-search.php>), “NIC” is the code for Cornell University. Field 950 is a local note field that shows this journal has a call number of “QA76.5 .E89” and was added to the Engineering Library on “7/30/81” (meaning July 30, 1981). Data from Field 008 and Field 040 inform my opinion that the library of Cornell University created an original MARC record for *Microprocessing and Microprogramming* on July 30, 1981, and added this journal to their Engineering Library, making the journal discoverable in the library catalog since then.

38. The Cornell MARC record (**Appendix 1005D**) is shorter than the Library of Congress MARC record (**Appendix 1005B**), but essential fields are the same in both records. Field 022 shows the ISSN of this journal is “0165-6074,” Field 245 shows the journal title, Field 260 shows the publisher information, Field 362

shows the journal began with “Vol. 7, no. 1 (Jan. 1981)” and Field 710 shows Euromicro is included to help users discover this journal. Subjects of this journal are represented in Field 950 by a Library of Congress Classification number of “QA76.5” that represents general works on digital computers (source: Library of Congress Classification - Schedule Q at https://www.loc.gov/aba/publications/FreeLCC/LCC_Q2020_TEXT.pdf) and by two Library of Congress subject headings in Fields 650, with *Microprocessors and Microprogramming* as the main topics, each followed by “Periodicals” as a form subdivision (encoded in subfield “x”) to show these topics are treated in a journal. The ISSN, title, publisher, beginning issue, the relationship between this journal and Euromicro, and subject representations in the Cornell MARC record are the same as those represented in the Library of Congress MARC record, confirming this record represents *Microprocessing and Microprogramming* that contains *Silva*.

39. This MARC record shows that *Microprocessor and Microprogramming* has been a long-running journal, and the MARC record makes the journal discoverable at the catalog of the library of Cornell University since 1981. Users would have been able to conduct subject searches to discover this journal. They would also have been able to search for the journal by title, ISSN and Euromicro.

40. The Cornell copy of *Silva* (**Appendix 1005C**) shows a date stamp of “AUG 25 1993.” The stamp has the appearance and distinctive characteristics of a typical check-in date stamp utilized by the library. As discussed earlier, libraries usually create one record for an entire journal, instead of creating records for individual issues of a journal, and they rely on a serial check-in system to monitor the receipt of new issues and update holdings information for users. Because journals publish latest research, the customary practice of academic libraries is to make a newly received issue available soon after serial check-in, usually within a week. It is therefore my opinion that the Cornell copy of the journal issue containing *Silva* would have been publicly accessible by September 1, 1993, at their Engineering Library.

F. Tulsa University Records

41. As discussed above, conference proceedings can be cataloged as serials or monographs. While many libraries cataloged the journal *Microprocessing and Microprogramming* as a serial, the library of the University of Tulsa cataloged the special issue on the EUROMICRO 93 conference as a monograph. I discovered their records by searching WorldCat for “Euromicro 93” for records, and found a record at <https://worldcat.org/title/226399526> that led me to their records for this conference proceedings. **Appendix 1005E** is a true and correct copy of the bibliographic and MARC records for “Proceedings EUROMICRO 93 Open System

Design Hardware, Software and Applications; Barcelona, September 6-9, 1993.” I personally located, identified and obtained these records. They are the type of records experts in my field would reasonably rely upon when forming their opinions. I used screen captures of the bibliographic and MARC records to produce **Appendix 1005E**.

42. Page 1 of **Appendix 1005E** is the bibliographic record that shows the conference name, in the authorized cataloging form, is included as the author, “Proceedings EUROMICRO 93 Open System Design Hardware, Software and Applications; Barcelona, September 6-9, 1993” is recorded as the title, and 1993 is the publication year. The series field shows this publication is “v.38, nos.1-5” of *Microprocessing and Microprogramming* and the second “Note” field shows it is “SHELVED AS VOL. 38, NOS. 1-5 OF THE PERIODICAL, ‘MICROPROCESSING AND MICROPROGRAMMING’.” The holdings field shows this work is shelved with “McFarlin Periodicals (Intermediate Level),” its call number is “Periodicals (see NOTE)” and its status is “Available.”

43. Page 2 is the MARC record. The first six digits of Field 008 shows the record was created on “930824” (*i.e.*, August 24, 1993), and the “s1993” code following the record creation date indicates this work has a single publication year of 1993. Field 040 shows “OKT” create the original MARC record. According to the *Directory of OCLC Members* (<https://www.oclc.org/en/contacts/libraries.html>),

“OKT” is the OCLC library symbol for the University of Tulsa. Data from Field 008 and Field 040 inform my opinion that the library of the University of Tulsa cataloged this volume of conference proceedings as a monograph and created an original record on August 24, 1993, making this work discoverable in their library catalog from then on.

44. The MARC record shows the conference name is used as the author and recorded in Field 111. Field 245 shows the conference title as the title, Fields 246 show two variant titles are included to help users discover this publication. Field 260 shows publisher information. Field 300 shows physical characteristics of this work. Two notes in Fields 500 clarify Euromicro 93 is the “Nineteenth EUROMICRO conference” and this work is volume 38, nos. 1-5 of the journal *Microprocessing and Microprogramming*. Field 710 shows Euromicro is included to help users discover this publication. Field 830 includes the journal title and volume and issue information to help users identify, discover and access this work.

45. Subjects of this work are represented by three Library of Congress subject headings in Fields 650, including “Computer network protocols |v Congresses,” “Computer network protocols,” and “Computer network protocols |x Standards |v Congresses” with “|v” indicating a form subdivision and “|x” indicating a topical subdivision. The “Congresses” form subdivision informs users that these

topics were discussed at a conference. Because periodicals in this library are shelved by journal title, no classification numbers are used to represent the journals.

46. This MARC record shows that the library of the University of Tulsa cataloged this volume of conference proceedings containing *Silva* as a monograph on August 24, 1993, making it discoverable in their catalog from that date on. Interested users would have been able to discover this work by the conference name, the conference title, variant titles, journal name, and Euromicro.

47. As discussed earlier, a customary library cataloging practice is to process a newly cataloged book quickly, usually within a week, to make it publicly accessible. It is therefore my opinion that this copy of the conference proceedings would have become publicly accessible by August 31, 1993, at the library of the University of Tulsa.

G. Usage Record

48. Actual usage of an article is reflected by other publications that make reference to it. The citation history of Google Scholar shows *Silva* was cited eight times, including two citations appeared in 1994: Ng, T. P., & Patel, V. N. (1994, October). Timely failure detection in a large distributed real-time system. In *Proceedings of Words' 94. The First Workshop on Object-Oriented Real-Time Dependable Systems* (pp. 118-123) and Brudło, P., & Krawczyk, H. (1994, December). A concept of a fully distributed processing strategy oriented for Unix

Local Area Networks. *Microprocessing and microprogramming*, 40(10-12), 723-726.

H. Summary of My Opinion on *Silva*

49. The date stamps on the Library of Congress copy of *Silva* (**Appendix 1005A**) and the Cornell copy of *Silva* (**Appendix 1005C**) show copies of the journal issue containing *Silva* were received by the Library of Congress on August 24, 1993, and by the library of Cornell University on August 25, 1993, respectively. It is therefore my opinion that the copies would have become publicly accessible by August 31, 1993, at the Library of Congress, and by September 1, 1993, at the library of Cornell University. The MARC record of the library of the University of Tulsa (**Appendix 1005E**) shows that on August 23, 1993, the library cataloged the volume of the EUROMICRO 93 conference proceedings containing *Silva*, which was published as vol. 38, nos. 1-5 of *Microprocessing and Microprogramming*, and made it discoverable in their catalog from that date on. The copy would have been publicly accessible by August 30, 1993, at that library. Citation history shows the earliest citation appeared in October of 1994.

III. Conclusion

50. I hereby declare that all statements made herein on my own knowledge are true and that all statements made on information and belief are believed to be

true, and further, that these statements were made with the knowledge that willful false statements and the like so made are made under penalty of perjury.

Date: 1-30-2023

Executed: Ingrid Hsieh-Yee
Ingrid Hsieh-Yee, Ph.D.

APPENDIX A

Ingrid Hsieh-Yee
Professor Emerita
Dept. of Library and Information Science
Catholic University of America
Washington, D.C. 20064
E-mail: hsiehyee@cua.edu

Education

Ph.D. 1990. Library and Information Studies, University of Wisconsin-Madison
Minors: Sociology and Psychology

M.A. 1984. Library and Information Studies, University of Wisconsin-Madison.

M.A. 1983. Comparative Literature, University of Wisconsin-Madison.

B.A. 1980. Foreign Languages and Literature, National Taiwan University.

Work Experience

Professor, School/Dept. of Library and Information Science, Catholic University of America,
1990-2022 (Assistant Professor, 1990-1996; Associate Professor, 1997-2004; Professor
2004-2022; Professor Emerita 2022)

Co-Chair, Dept. of Library and Information Science, Catholic University of America, June 2015-
August 2016.

Acting Dean, School of Library and Information Science, Catholic University of America,
January 2010-June 2012.

Cataloger, Dept. of Legislative Reference Library, Annapolis, Maryland, 1989-1990.

Lecturer, School of Library and Information Studies, University of Wisconsin-Madison, 1988.

Teaching Assistant, School of Library and Information Studies, University of Wisconsin-
Madison, 1986-1988.

Cataloger, Health Sciences Library, University of Wisconsin-Madison, 1984-1986.

Areas of Teaching and Research Interests

Information Organization and Access; Metadata; Cataloging & Classification; Information
Architecture; Information Retrieval; Digital Collections; Social Media; Scholarly

Communication; Information Behavior; Health Informatics; Human Computer Interaction; Usability Studies

Grants & Honors

Cultural Heritage Information Management Project. IMLS grant. Amount: \$498,741. Period: Aug. 2012 to July 2015. Co-PI with Dr. Youngok Choi.

D.C. Health Information Technology (HIT4): Building Capacity & Providing Access in Our Nation's Capital. Dept. of Labor H2B Training Grant. Grant amount: \$4,175,500. Grant period: Nov. 2011 to Dec. 2015. Partner with the Metropolitan School of Professional Studies of the Catholic University of America, Children's National Medical Center, D.C. Department of Employment Services, Holy Cross Hospital, Howard University, Center for Urban Progress, Providence Hospital, and Sibley Memorial Hospital.

Capital Health Careers Project. Department of Labor Healthcare Sector and Other High Growth and Emerging Industries Grant. Grant amount: \$4,953,999. Grant period: March 2010 – February 2013. Awarded to a group of healthcare organizations and educational institutions in Washington, D.C. Providence Health Foundation of Providence Hospital (Lead institution). Part of the grant supported the development of a Master's degree program in Information Technology with a concentration in Health Information Technology offered by the School of Library and Information Science.

The Washington D.C. School Librarians Project. IMLS grant. Grant amount: \$412,660. Grant period: Aug. 2007 – June 2011. The School partnered with the District of Columbia Public Schools (DCPS) and the District of Columbia Library Association to educate and mentor school media specialists for the DCPS system. PI, Jan. 2010 to June 2011.

SIG Member of the Year, American Society for Information Science and Technology (2009).

Most Outstanding Paper of *OCLC Systems & Services* (2001).

ALISE Research Grant (2001).

Most Outstanding Paper of *OCLC Systems & Services* (2000).

Research Grant from ERIC (1999-2000).

Best Research Paper Award; Association for Library and Information Science Education (1998).

Research Grants, Catholic University of America. 1991, 1992, 1993, 1996, 1998, 1999, 2004, 2005, 2006, 2007, 2013-14.

Cooperative Faculty Research Grant, Consortium of Universities in the Washington Metropolitan Area (1993-1994).

Cooperative Research Grant, Council on Library Resources (1993-1994).

Journal of the American Society for Information Science Best Paper Award (1993).

ASIS/ISI Information Science Doctoral Dissertation Scholarship (1989).

HEA Title IIB Fellowship (Dept. of Education) (1989)

Chinese-American Librarians Association Scholarship (1987).

Beta Phi Mu (1985).

Vilas Fellowship, University of Wisconsin-Madison. 1984

Publications

Hsieh-Yee, I. (2021). Can We Trust Social Media? *Internet Reference Services Quarterly*, 1-15.
Published online: 28 July, 2021. 10.1080/10875301.2021.1947433.

Bailey, T., & Hsieh-Yee, I. (2020). Combating the Sharing of False Information: History, Framework & Literacy Strategies, *Internet Reference Services Quarterly*, 24 (1-2), 9-30.

Choi, Y., and Hsieh-Yee, I. (2010). Finding Images in an OPAC: Analysis of User Queries, Subject Headings, and Description Notes. *Canadian Journal of Information and Library Science*, 34(3): 271 – 295.

Hsieh-Yee, I. (2008). Educating Cataloging Professionals in a Changing Information Environment. *Journal of Education for Library and Information Science*, 46(2): 93-106.

Vellucci, S. L., Hsieh-Yee, I., and Moen, W.E. (2007). The Metadata Education and Research Information Commons (MERIC): A Collaborative Teaching and Research Initiative. *Education for Information*, 25(3&4): 169-178.

NISO Framework for Guidelines for Building Good Digital Collections. 3rd ed. Baltimore, MD: National Information Standards Organization, 2007. Also available online: <http://www.niso.org/framework/framework3.pdf> (NISO Working Group members: Priscilla Caplan (chair), Grace Agnew, Murtha Baca, Tony Gill, Carl Fleischhauer, Ingrid Hsieh-Yee, Jill Koelling, and Christie Stephenson.)

Choi, Y., Hsieh-Yee, I., and Kules, B. (2007). Retrieval Effectiveness of TOC and LCSH. *Proceedings of the Joint Conference on Digital Libraries*, pp. 233-234.

Vellucci, S., and Hsieh-Yee, I. (2007). They Didn't Teach Me That in Library School! Building

a Digital Teaching Commons to Enhance Metadata Teaching, Learning and Research. *Proceedings of the National Conference of the Association of College and Research Libraries, Baltimore, MD*, pp. 26-31.

Mitchell, Vanessa, and Ingrid Hsieh-Yee. (2007). Converting Ulrich's Subject Headings to FAST Headings: A Feasibility Study. *Cataloging & Classification Quarterly*, 45(1): 59-85.

Hsieh-Yee, I., Tang, R., and Zhang, S. (2007). User Perceptions of a Federated Search System. *IEEE Technical Committee on Digital Libraries Bulletin*, Summer 3(2) (URL = <http://www.ieee-tcdl.org>).

Tang, R., Hsieh-Yee, I., and Zhang, S. (2007). User Perceptions of MetaLib Combined Search: An Investigation of How Users Make Sense of Federated Searching." *Internet Reference Services Quarterly*, 12(12): 211-236.

Hsieh-Yee, I., Tang, R., and Zhang, S. (2006). User Perceptions of a Federated Search System. *Proceedings of the Joint Conference on Digital Libraries, June 11-15, 2006, Chapel Hill*, p. 338.

Hsieh-Yee, I. (2006). *Organizing Audiovisual and Electronic Resources for Access: A Cataloging Guide*. 2nd ed. Westport, Conn.: Libraries Unlimited.

NISO A Framework of Guidance for Building Good Digital Collections. 2nd ed. Bethesda, MD: National Information Standards Organization, 2004. Framework Advisory Group: Grace Agnew, Liz Bishoff, Priscilla Caplan (Chair), Rebecca Gunther and Ingrid Hsieh-Yee.

Hsieh-Yee, I. (2004). Cataloging and Metadata Education in North American LIS Programs. *Library Resources & Technical Services*, 48(1): 59-68.

Hsieh-Yee, I. (2004). Cataloging and Metadata Education. In Gary E. Gorman (Ed.), *International Yearbook of Library and Information Management 2003: Metadata Applications and Management*, (pp.204-234). London: Facet Publishing.

Yee, P. L., Hsieh-Yee, I., Pierce, G.R., Grome, R., and Schantz, L. (2004). Self-Evaluative Intrusive Thoughts Impede Successful Searching on the Internet. *Computers in Human Behavior*, 20(1): 85-101.

Hsieh-Yee, I. (2003). Cataloging and Metadata Education: A Proposal for Preparing Cataloging Professionals of the 21st Century. A report submitted to the ALCTS-Education Task Force in response to Action Item 5.1 of the *Bibliographic control of Web Resources: A Library of Congress Action Plan*. Approved by the Association for Library Collections and Technical Services. Web version available since April 2003 at <http://lcweb.loc.gov/catdir/bibcontrol/CatalogingandMetadataEducation.pdf>.

- Hsieh-Yee, I. (2002). Cataloging and Metadata Education: Asserting a Central Role in Information Organization. *Cataloging & Classification Quarterly* 34(½): 203-222.
- Hsieh-Yee, I., and Smith, M. (2001). The CORC Experience: Survey of Founding Libraries, Part I. *OCLC Systems & Services*, 17: 133-140. (Received "The Most Outstanding Paper of OCLC Systems & Services in 2001" award.)
- Hsieh-Yee, I., and Smith, M. (2001). The CORC Experience: Survey of Founding Libraries, Part II, Automated Tools and Usage. *OCLC Systems & Services*, 17: 166-177. (Received "The Most Outstanding Paper of OCLC Systems & Services in 2001" award.)
- Hsieh-Yee, I. (2001). ERIC User Services: Changes and Evaluation for the Future. *Government Information Quarterly*, 18: 31-42.
- Hsieh-Yee, Ingrid. (2001). Research on Web Search Behavior. *Library and Information Science Research*, 23: 167-185.
- Logan, E., and Hsieh-Yee, I. (2001). Library and Information Science Education in the Nineties. *Annual Review of Information Science and Technology*, 35: 425-477.
- Hsieh-Yee, I. (Ed.) (2001). *Library and Information Science Research*, 23 (2). A special issue in honor of the retirement of Douglas L. Zweizig.
- Hsieh-Yee, I. (2000). *ERIC User Services: Evaluation in a Decentralized Environment*. Washington, D.C.: Dept. of Education.
- Hsieh-Yee, Ingrid. (2000). *Organizing Audiovisual and Electronic Resources for Access: A Cataloging Guide*. Littleton, CO: Libraries Unlimited.
- Hsieh-Yee, I. (2000). Organizing Internet Resources: Teaching Cataloging Standards and Beyond. *OCLC Systems & Services*, 16: 130-143. (Received "The Most Outstanding Paper of OCLC Systems & Services in 2000" award.)
- Hsieh-Yee, I. (1998). The Retrieval Power of Selected Search Engines: How Well Do They Address General Reference Questions and Subject Questions? *Reference Librarian*, 60: 27-47.
- Hsieh-Yee, I. (1998). Search Tactics of Web Users in Searching for Texts, Graphics, Known Items and Subjects: A Search Simulation Study. *Reference Librarian*, 60: 61-85. (Received the 1997 Best ALISE Research Paper Award.)
- Hsieh-Yee, I. (1997). Access to OCLC and Internet Resources: LIS Educators' Views and Teaching Practices. *RQ*, 36: 569-86.

- Hsieh-Yee, I. (1997). Teaching Online and CD-ROM Resources: LIS Educators' Views and Practices. *Journal of Education for Library and Information Science*, 38: 14-34.
- Hsieh-Yee, I. (1996). The Cataloging Practices of Special Libraries and Their Relationship with OCLC. *Special Libraries*, 87: 10-20.
- Hsieh-Yee, I. (1996). Modifying Cataloging Practice and OCLC Infrastructure for Effective Organization of Internet Resources. In *Proceedings of the OCLC Internet Cataloging Colloquium*. [Online]. Available: <http://www.oclc.org/oclc/man/colloq/hsieh.htm>
- Hsieh-Yee, I. (1996). Student Use of Online Catalogs and Other Information Channels. *College & Research Libraries*, 57: 161-175.
- Hsieh-Yee, I. (1995). Ten entries in James S. C. Hu (Ed.), *Encyclopedia of Library & Information Science*, 913, 1028-29, 1036, 1037, 1145-46, 1514, 1575, 1763-64, 2216-27, 2378-79. Taipei, Taiwan: Sino-American Publishing. (Topics include "Advanced Technology/Libraries," "Information Ethics," "Instruction on Cataloging and Classification," "Instruction on Reference Services.")
- Hsieh-Yee, I. (1993). Effects of Search Experience and Subject Knowledge on Online Search Behavior: Measuring the Search Tactics of Novice and Experienced Searchers. *Journal of the American Society for Information Science*, 44: 161-174. (Received the 1993 Best JASIS Paper Award.)

Works in Progress

- Hsieh-Yee, I. Actionable lessons for debunking false information: A comparative analysis of literacy lessons in communication and library and information science
- Hsieh-Yee, I. Representing diverse voices and communities in cataloging and indexing: The evolution of controlled vocabularies
- Hsieh-Yee, I. Information Organization and Retrieval in Practice (a monograph scheduled for completion and publication in 2023)

Presentations

- Bailey, T. & Hsieh-Yee, I. (February 2020). The Phenomena of Sharing Misinformation and the Need for Information Literacy. Presented at the 2020 Bridging the Spectrum Symposium, Washington, D.C., February 14, 2020.

- Hsieh-Yee, I. and Fragan-Fly, J. (May 2018) *Trends, Design & Strategies for Digital Scholarship Services*. Presented at the 2018 Maryland/Delaware Library Association Conference, Cambridge, MD.
- Hsieh-Yee, I. (February, 2018) *Research Data Management: What It Takes to Succeed*. Presented at the 10th Bridging the Spectrum Symposium, Washington, D.C.
- Hsieh-Yee, I. (February, 2017) *Research Data Management: New Competencies and Opportunities for Information Professionals*. Presented at the 9th Bridging the Spectrum Symposium, Washington, D.C.
- Hsieh-Yee, I. and Lawton, P. (February, 2017) *Enhancing Catholic Portal Searches with User Terms and LCSH*. Presented at the 9th Bridging the Spectrum Symposium, Washington, D.C.
- Hsieh-Yee, I. (2016, October) *Visualizing Data for Information*. Presented at the 2016 Virginia Library Association Conference, Hot Springs, VA.
- Hsieh-Yee, I. (2016, August) *Religious Materials Toolbox for Archivists: Solutions to Problems Facing the Profession*. Presented at Archives * Records 2016, Atlanta, GA.
- Hsieh-Yee, I. and Lawton, P. (2016, March) *Enhancing Retrieval of Catholic Materials with LCSH Knowledge Structure*. Presented at the 2016 Catholic Library Association Conference, San Diego, CA.
- Fagan-Fry, J. and Hsieh-Yee, I. (2016, February) *Approaches to Digital Scholarship at Top Universities around the World: Scholarly Publishing in the Digital Age*. Presented at the 8th Bridging the Spectrum Symposium, Washington, D.C.
- Hsieh-Yee and Fagan-Fry, J. (2016, January) *Innovative Services for Digital Scholarship at Top 100 Research Libraries of the World*. Poster presented at the 2016 Annual Conference of the Association for Library and Information Science Education, Boston, Mass.
- Hsieh-Yee, I. and Lawton, P. (2015, June). *Crowdsourcing terms for CRRA portal themes*. Poster presented at the third CRRA symposium and annual meeting, Bringing the created toward the Creator: Liturgical art and design since Vatican II. Catholic Theological Union, Chicago, Illinois.
- Hsieh-Yee, I. and Lawton, P. (2015, February). *Crowdsourcing terms for thematic exploration in the Catholic Portal*. Poster presented at the 7th Annual Bridging the Spectrum Symposium, Washington, D.C.
- Hsieh-Yee, I., James, R., and Fagan-Fry, J. (2015, February). *Support for digital scholarship at top university libraries of the world*. Poster presented at the 7th Annual Bridging the Spectrum Symposium, Washington, D.C.

- Hsieh-Yee, I., Zhang, S., Lin, K., and Cherry, S. (2015, February). *Thus said the end users: Summon experience and support for research workflows*. Poster presented at the 7th Annual Bridging the Spectrum Symposium, Washington, D.C.
- Yontz, E., Hsieh-Yee, I., & Houston, S. (2015, February). *Healthy Heroes Summer Reading Club: Developing healthy youth at public libraries*. 11th Annual Jean Mills Health Symposium, Greenville, North Carolina.
- Yontz, E., Hsieh-Yee, I., and Houston, S. (2015, January). *Healthy youth and libraries: A pilot study*. Association for Library & Information Science Education (ALISE) Annual Conference, Chicago, Illinois.
- Hsieh-Yee, I. (2014, May). *Linking CRRA resources to portal themes via authority files*. Presented at the Catholic Research Resources Alliance 2014 Membership Meeting, Marquette, WI.
- Hsieh-Yee, I. (2014, April). *Enhancing subject access to CRRA resources*. Presented at the 2014 Catholic Library Association Conference, Pittsburgh, PA.
- Hsieh-Yee, I. (2014, January). *Health Information Technology Program: Educational entrepreneurship in action*. Presented at the 2014 annual Conference of the Association for Library and Information Science Education, Philadelphia, PA.
- Hsieh-Yee, I., Zhang, S., Lin, K., and Cherry, S. (2014, January). *Discovering information through Summon: An analysis of user search strategies and search success*. Paper presented at the 6th Bridging the Spectrum Symposium, Washington, D.C.
- Hsieh-Yee, I. (2012, December). *National Digital Stewardship Alliance and SLIS at CUA: An Educational Partnership*. Paper presented at Best Practices Exchange: Acquiring, Preserving, and Providing Access to Government Information in the Digital Era, Annapolis, MD.
- Choi, Y. and Hsieh-Yee, I. (2010, January). *Finding Images in an OPAC: Analysis of User Queries, Subject Headings, and Description Notes*. Paper presented at 2nd Annual Bridging the Spectrum Symposium, Catholic University of America, Washington, D.C.
- Hsieh-Yee, I. and Coogan, J. (2010, January). *Google Scholar vs. Academic Search Premier: What Libraries and Searchers Need to Know*. Paper presented at 2nd Annual Bridging the Spectrum Symposium, Catholic University of America, Washington, D.C.
- Hsieh-Yee, I. (2009, November). *Information Science Education: An LIS School's Perspective*. Paper presented at Annual Meeting of the American Society for Information Science and Technology, Vancouver, British Columbia, Canada.

- Hsieh-Yee, I., Menard, E., Ya-Ning Chen, A., Shu-Jiun Chen, S., Kalfatovic, M. R., Wisser, K. M. (2009, November). *Information Organization in Libraries, Archives and Museums: Converging Practices and Collaboration Opportunities*. Presented at Annual Meeting of the American Society for Information Science and Technology, Vancouver, British Columbia, Canada. (Organizer and moderator of this panel.)
- Hsieh-Yee, I. and Coogan, J. (2009, July). *Catching up to Google Scholar: The Retrieval Power of Academic Search Premier and Google Scholar*. Poster presented at American Library Association Conference, Chicago, Illinois.
- Hsieh-Yee, I., with the CUA Scholarly Communications Project Team. (2009, January). *Digital Scholarship@CUA: Developing an Institutional Repository for CUA*. Poster presented at 1st Annual Bridging the Spectrum Symposium, Catholic University of America, Washington, D.C.
- Wise, M., Cylke, K., and Hsieh-Yee, I. (2009, January). *Digital Talking Books: Meeting the Needs of the Blind and the Handicapped*. Paper presented at the Bridging the Spectrum Symposium, Catholic University of America, Washington, D.C.
- Hsieh-Yee, I. (2009, January). *User Expectations of MERIC*. Presented at the Information Organization Competencies for the 21st Century Discussion Session of the 2009 Conference of the Association for Library and Information Science Education, Denver, Colorado.
- Choi, Y., and Hsieh-Yee, I. (2008, November). *Subject Access for Images in an OPAC*. Annual Meeting of the American Society for Information Science and Technology, Columbus, Ohio. (Also co-organized a panel on Retrieving and Using Visual Resources: Challenges and Opportunities for Research and Education.)
- Hsieh-Yee, I. (2008, June). *Educating Cataloging Professionals in a Changing Information Environment*. National Taiwan University, Taipei, Taiwan.
- Vellucci, S. L., Moen, W.E., Hsieh-Yee, I., Marson, B., and Wisser, K. (2008, January) *Building a Metadata Education and Research Community through MERIC (Metadata Education and Research Information Commons): Demo and Stakeholder Input*. A panel presented at the 2008 Conference of the Association for Library and Information Science Education, Philadelphia, Pennsylvania.
- Hsieh-Yee, I., Choi, Y. and Kules, B. (2007, October). *Searching for Books and Images in OPAC: Effects of LCSH, TOC and Subject Domains*. A poster presented at the American Society for Information Science and Technology Annual Meeting, Milwaukee, Wisconsin.
- Hsieh-Yee, I. and Coogan, J. (2007, August) *Google Scholar vs. Academic Search Premier: A Comparative Analysis*. Presented to the Faculty and Staff of the University of the District

of Columbia.

- Hsieh-Yee, I. and Coogan, J. (2007, June). *Google Scholar vs. Academic Search Premier: A Comparative Analysis*. Presented to the Washington Research Library Consortium Community, Catholic University of America, Washington, D.C.
- Hsieh-Yee, I., Choi, Y., and Kules, B.. (2007, June). *What Users Need for Subject Access: Table of Contents or Subject Headings?* A poster presented at the 2007 American Library Association Annual Conference, Washington, D.C., June 2007.
- Choi, Y., Hsieh-Yee, I., and Kules, B. (2007, June). *Retrieval Effectiveness of TOC and LCSH*. A paper presented at the Joint Conference on Digital Libraries 2007, Vancouver, Canada.
- Vellucci, S. L., Hsieh-Yee, I., and Moen, W.E. (2007, May). *If We Build It, Will They Come? Building a Community of Practice for Metadata Stakeholders*. A poster presented at the Rutgers University Research Day, Bridgeton, New Jersey.
- Hsieh-Yee, I. (2007, May). *Federated Searching: User Experience & Perceptions*. International Conference on Information Organization & Retrieval, National Taiwan University, Taipei, Taiwan.
- Hsieh-Yee, I. (2007, May). *Search Performance of Google Scholar and Academic Search Premier*. International Conference on Information Organization & Retrieval, National Taiwan University, Taipei, Taiwan.
- Hsieh-Yee, I. (2007, May) *MERIC: Building a Digital Commons for Metadata Education & Research*. International Conference on Information Organization & Retrieval, National Taiwan University, Taipei, Taiwan.
- Hsieh-Yee, I., and Coogan, J. (2007, March/April). *A Comparative Analysis of Google Scholar and Academic Search Premier*. Poster presented at the Association of College & Research Libraries 13th National Conference, Baltimore, Maryland.
- Vellucci, S. L. and Hsieh-Yee, I. (2007, March/April) *They Didn't Teach Me That in Library School! Building a Digital Teaching Commons to Enhance Metadata Teaching, Learning and Research*. On-site presentation and Webcast by Elluminate. A contributed paper presented at the Association of College & Research Libraries 13th National Conference, Baltimore, Maryland. The acceptance rate for contributed paper was 20%. This paper was one of 10 conference papers chosen for live webcast during the conference.
- Moen, W., Hsieh-Yee, I. and Vellucci, S.L. (2007, January) *A DSpace Foundation for a Teaching & Research Commons: The Metadata Education and Research Information Commons*. A poster session presented at the Open Repositories Conference 2007, San Antonio, Texas.

- Tang, R., Hsieh-Yee, I., and Zhang, S. (2006, November) *User Perception of MetaLib Combined Search*. Paper presented at the Annual Meeting of the American Society for Information Science and Technology, Austin, Texas, Nov. 2006.
- Hsieh-Yee, I. (2006, November). *Federated Searching: User Perceptions, System Design, and Library Instructions*. Paper presented at the Annual Meeting of the American Society for Information Science and Technology, Austin, Texas. (Panel organizer, moderator, presenter).
- Hsieh-Yee, I. (2006, November). *Building a Digital Teaching Commons to Enhance Teaching and Learning: The MERIC Experience and Challenges*. Paper presented at the Annual Meeting of the American Society for Information Science and Technology, Austin, Texas. (Panel organizer, moderator, presenter)
- Hsieh-Yee, I. (2006, September). *Search Performance of Google Scholar and Academic Search Premier*. Paper presented at the ERIC Publishers Meeting, Washington, D.C.
- Hsieh-Yee, I., Zhang, S., and Rong Tang, R. (2006, June). *User Perceptions of a Federated Search System*. Poster presented at Joint Conference on Digital Libraries, Chapel Hill, North Carolina.
- Hsieh-Yee, I. and Zhang, S. (2006, June). *Preparing Users for Federated Search: Implications of a MetaLib User Perceptions Study*. Paper presented at the 2006 Ex Libris User Groups of North America Conference, Knoxville, Tennessee.
- Hsieh-Yee, I. (2006, January). *MERIC Organizations and Navigation*. Paper presented at the 2006 ALISE Annual Conference, San Antonio, Texas.
- Hsieh-Yee, I. (2006, January). *Metadata and Cataloging Education: Recommended Competencies*. Paper presented at the 2006 ALISE Annual Conference, San Antonio, Texas.
- Hsieh-Yee, I. (2005, November). *Digital Library Evaluation: Progress & Next Steps*. Presentation at the Annual Meeting of the American Society for Information Science & Technology, Charlotte, North Carolina.
- Hsieh-Yee, I. (2005, August). *Providing Access to Digital Content: Issues for DL Managers*. Presentation at MDK12 Digital Library Steering Committee Meeting, Columbia, Maryland.
- Hsieh-Yee, I. (2005, April). *Enhancing Teaching and Learning: The Role of School Library Media Specialists*. Presentation at Meeting of the Baltimore County Public School System School Media Specialists, Baltimore, Maryland.

- Hsieh-Yee, I. (2005, January). *Subject Access and Users: Insights & Inspirations from Marcia J. Bates*. Paper presented at the Historical Perspectives SIG, 2005 Conference of the Association for Library and Information Science Education, Boston, Massachusetts.
- Hsieh-Yee, I. (2005, January). *Electronic Resource Management: Practice, Employer Expectations, & CE Interests*. Paper presented at Technical Services Education SIG, 2005 Conference of the Association for Library and Information Science Education, Boston, Massachusetts.
- Hsieh-Yee, I. (2004, October). *Library Professionals for the Digital Age: Competencies & Preparation*. Paper presented at Bibliographic Access Management Team meeting, Library of Congress, Washington, D.C.
- Hsieh-Yee, I. (2004, January). *Cataloging and metadata expertise for the digital era*. Presented at Preparing 21st Century Cataloging and Metadata Professionals: A Workshop for Educators and Trainers, San Diego and sponsored by ALCTS, ALISE, LC, and OCLC.
- Hsieh-Yee, I. (2004, January). *Educating catalogers for the digital era*. Paper presented at the Technical Services SIG, 2004 Conference of the Association for Library and Information Science Education, San Diego.
- Hsieh-Yee, I. (2003, July). *Cataloging Education for the 21st Century*. A presentation at the Library of Congress, Washington, D.C.
- Hsieh-Yee, I. (2002, January) *Metadata Education and Research Priorities: A Delphi Study of Metadata Experts*. Presentation at the 2002 Conference of the Association for Library and Information Science Education, New Orleans.
- Hsieh-Yee, I. (2001, November). *A Delphi Study of Metadata: Preliminary Findings*. Poster session at the 2001 Annual Meeting of the American Society for Information Science & Technology, Washington, D.C.
- Hsieh-Yee, I. (2001, June). *Resources on Asian American Children: Analysis of Retrieval by Search Engines and WorldCat*. Presentation at the National Conference on Asian Pacific American Librarians, San Francisco.
- Hsieh-Yee, I. (2001, January). *Delphi Study on Metadata: Project Design*. Presentation at Research Awards Session, Association for Library & Information Science Education, Washington, D.C.
- Hsieh-Yee, I. (2000, May). *Web Search Behavior Research: Progress and Implications*. Presentation at the Symposium on Evaluating Library and Information Science Research, University of Wisconsin-Madison, Madison, Wisconsin.

- Hsieh-Yee, I. (2000, March). *ERIC User Services: Evaluation in a Decentralized Environment*. Presentation at the National ERIC Joint Directors/Technical Meeting, Arlington, Virginia.
- Hsieh-Yee, I. (2000, January). *Enhancing Learning with Web Technology*. Presentation at Faculty Conversations, Catholic University of America, Washington, D.C.
- Hsieh-Yee, I. (2000, January). *From Surrogates to Objects: CUA's Approaches to Organizing Electronic Resources*. Paper presentation at the Annual Conference of the Association for Library and Information Science Education, San Antonio, Texas.
- Yee, P., and Hsieh-Yee, I. (1997, November). *Individual Differences in Search Behavior on the WWW*. A poster session presented at the 38th Annual Meeting of the Psychonomic Society, Philadelphia, Pennsylvania.
- Hsieh-Yee, I. (1997, April). *Research + Marketing + Preparation = Job!* Presented at the "Workshop on Resume and Interview Techniques," Special Libraries Association, Student Chapter, Catholic University of America, Washington, D.C.
- Hsieh-Yee, I. (1997, February). *Creating CyberCatalogers: Education and Training*. Presentation at ALA's Midwinter Meeting, Washington, D.C.
- Hsieh-Yee, I. (1997, February). *Search Tactics of Web Users in Searching for Texts, Graphics, Known Items and Subjects: A Search Simulation Study*. Presented at the Conference of the Association for Library and Information Science Education, Washington, D.C.
- Hsieh-Yee, Ingrid. "Beginning Your Special Library/Information Center Career." Presented at SLA's "Career Day," Jan. 11, 1997, Catholic University of America.
- Hsieh-Yee, I. (1996, September). *The Roles of Library and Information Scientists in Managing Electronic Information*. Presentation at Hamilton College, Clinton, New York.
- Hsieh-Yee, I. (1996, May). *The Future of Cataloging as a Profession*. Presented at "The Cataloging Forum, Library of Congress, Washington, D.C.
- Hsieh-Yee, I. (1994, October). *The Impact of the Internet on OPACs*. Presented at the Third Workshop on User Interfaces for OPACs, Library of Congress, Washington, D.C.

Reports

- Hsieh-Yee, I., with Knowledge Management Competencies and Performance Action Group of the Federal Knowledge Management Initiative. "From Knowledge Management Competencies to Improved Organizational Performance." April 9, 2009.

- Hsieh-Yee, I., with Knowledge Practices Action Group of the Federal Knowledge Management Initiative. "KM Practice in Government Agencies: Findings and Recommendations." April 9, 2009.
- Hsieh-Yee, I. "Delphi Study on Metadata." 2001. Three quarterly reports submitted to the Association for Library and Information Science Education.
- Hsieh-Yee, I. "College Students' Information Channels: Patterns of Use and Possible Factors in Channel Selection." 1995. Submitted to the Catholic University of America.
- Hsieh-Yee, I. "The Information-Seeking Patterns of Scholars and Their Use of an Online Information System." 1994. Submitted to the Council on Library Resources.

Book Reviews

- Review of *The Measurement and Evaluation of Library Services*, by Sharon L. Baker and F. Wilfrid Lancaster. *Information Processing and Management* 30 (1994): 450-52.
- Review of *Subject Access to Films and Videos*, by Sheila S. Intner and William E. Studwell; and *Cataloging Unpublished Nonprint Materials*, by Verna Urbanski with Bao Chu Chang and Bernard L. Karon. *Information Processing and Management* 30 (1994): 449-50.
- Review of *Automated Information Retrieval in Libraries: A Management Handbook*, by Vicki Anders. *Journal of Library and Information Science* 19 (1993): 98-100.
- Review of *Full Text Databases*, by Carol Tenopir and Jung Soon Ro. *Information Processing and Management* 28 (1992): 667-68.
- Review of *Descriptive Cataloging for the AACR2R And USMARC: A How-to-Do It Workbook*, by Larry Millsap and Terry Ellen Ferl. *Information Processing and Management* 28 (1992): 809-11.
- Review of *MARC Manual: Understanding and Using MARC Records*, by Deborah J. Byrne. *Information Processing and Management* 28 (1992): 537-38.

Service

Professional Associations and Societies

- Library of Congress. RDA Training Program for the Profession. Co-authored with Tim Carlton. 2013-2014.
- 2014 Digital Preservation Outreach & Education Survey. Contributed to the design of the survey, 2014.
- National Digital Stewardship Alliance. Outreach Committee. 2011-2014.
- National Digital Stewardship Residency Program. Advisory Group, 2012-2013.
- FEDLINK Health Information Technology Advisory Council, 2011-2015.

- 2012 Joint Conference on Digital Libraries. Program Planning Committee, Pre-Conference Proposals Review Committee, 2012
- Catholic Research Resources Alliance. Five-Year Strategic Plan Task Force, 2011-2012
- Institute of Museum and Library Services. Grant reviewer. 2004, 2005, 2010.
- Association for Library and Information Science Education.
 - * ALISE Bodan Wynar Research Paper Award Committee, 2015, 2016, 2017
 - * ALISE Eugene Garfield Dissertation Award Competition, Jury, 2013, 2014
 - * ALISE Research Grant Competition Committee. Chair, 2012
 - * Pratt-Severn Faculty Innovation Award. Chair, 2009, 2010
 - * ALISE Doctoral Poster Jury, 2012
 - * “Information Organization Competencies for the 21st Century” Discussion session leader. 2009 Conference of the Association for Library and Information Science Education.
 - * Assisted Technical Services SIG Convener in organizing a program, ““Building a Metadata Education and Research Community through MERIC (Metadata Education and Research Information Commons): Demo and Stakeholder Input” for the 2008 ALISE conference.
 - * Association for Library Collections and Technical Services/Association for Library and Information Science Education (ALCTS/ALISE) Metadata Education and Research Information Center (MERIC) Advisory Board, Co-Chair (with Sherry Vellucci), 2005-2007. Chair, 2008-2009 (leading the effort to build MERIC, a repository and collaborative space for metadata educators, practitioners, and researchers)
 - * Technical Services SIG, Convener, 2004-2005. Organized a program on “Electronic Resources Management: Current Practices, Employer Expectations, and Teaching Strategies” for the 2005 conference in Boston, Massachusetts.
 - * Technical Services SIG, Convener, 2003-2004. Organized a program on “Organizing Information with Metadata: Desired Competencies and Teaching Innovations” for the 2004 conference.
 - * Technical Services SIG, Convener, 1999-2000. Organized a program on "Teaching the Organization of Electronic Resources" for the 2000 conference.
 - * Curriculum SIG, Co-convener (with Sibyl Moses), 1996-97. Organized a program on “Government Information Policy” for the 1997 conference.
- American Society for Information Science & Technology.
 - * Reviewer, Conference program panel submissions and poster submissions, 2005, 2006, 2007, 2009, 2011, 2012, 2013, 2014, 2015, 2016, 2017
 - * Nomination Committee, 2009-2011
 - * Information Science Education Special Interest Group. American Society for Information Science and Technology. Chair-Elect, 2007-2008. Chair 2008-2009.
 - * Committee on Information Science Education. 1999-2006.
 - * Committee on Information Science Education. Organizing Committee for an orientation program for students at ASIS annual meetings, 1999-2001
 - * Committee on Information Science Education. Sub-committee on Student Welfare (focusing on issues related to master's education), 1998-2001

- * SIG ED. Organizing Committee for the "Seminar on Research and Career Development" for junior researchers. 1995-96 (chair), 1997-2001
- * ISI Doctoral Dissertation Proposal Scholarship Jury, 1997; 2001, 2002
- * Pratt-Severn Best Student Research Paper Award Jury. Chair. 1997
- * 1998 Midyear Meeting (referee of contributed papers), 1997
- * Organizer and moderator of the ASIS Doctoral Forum and the Doctoral Research Seminar 1994-1995
- * SIG Human Computer Interaction. Chair-Elect, Chair, 1993-1995
- * Doctoral Forum Award Jury, 1995
- * Best Student Paper Award Jury, 1995
- American Library Association.
 - * Committee on Accreditation, External Review Panelist, 2009- (site visiting team 2013-2014; site visiting team 2016-2017)
 - * Association for Library Collections and Technical Services Task Force on Competencies and Education for a Career in Cataloging, member, 2008-2009
 - * Facilitator for "What They Don't Teach in Library School: Competencies, Education and Employer Expectations for a Career in Cataloging," an Association for Library Collections and Technical Services Preconference, June 22, 2007 in Washington, D.C. Also a local liaison for bringing this program to the Catholic University of America.
 - * Facilitator for a discussion on "Effect of Electronic Resources on Technical Services" at ALA's Midwinter Meeting held in Feb. 1997 in Washington, D.C.
 - * International Relations Committee, Subcommittee Task Force for IFLA and China, 1994-1997
- Virginia Association of School Librarians. Scholarships and Awards Committee. 2010-2012
- Federal Knowledge Management Initiative, Knowledge Management Practices Action Group. Member. 2009 (leading the effort to build a knowledge management repository)
- Federal Knowledge Management Initiative, Knowledge Management Competencies & Learning Action Group. Member. 2009 (developing an action plan for helping government knowledge workers and government agencies to develop knowledge management competencies)
- National Center for Education Statistics. Technical Review Panel. 2008.
- External evaluator for a case of promotion to full professorship. University of Tennessee. 2008.
- National Information Standards Organization (NISO). Advisory Board, Revision of "IMLS Framework of Guidance for Building Good Digital Collections," 2004, 2007.
- Library of Congress, Bibliographic Control of Web Resources: A Library of Congress Action Plan. Principal Investigator of Action Item 5.1, focusing on cataloging and metadata education for students and new librarians, 2002-2003. (worked with the Association for Library Collections and Technical Services, Education Task Force)

- Chinese American Librarians Association
 - * Chinese American Librarians Association Outstanding Library Leadership Award in Memory of Dr. Margaret Chang Fung, Award Committee, 2016-2017
 - * Achievement Award Jury, 2000-2001
 - * CALA Goal 2000 Task Force, 1997
 - * Scholarship Committee, 1995, 1996-1997 (chair)
 - * Board of Directors, 1994-1997
 - * Publication Committee, 1993-1995
 - * International Relations Committee, 1993-1996
- SailorSM Assessment Advisory Group (An impact study of Sailor, Maryland's Public Information Network), 1995
- Editorial boards
 - Journal of Library & Information Science. Editorial Board, 2012-
 - Chinese American Librarians Association, *Occasional Papers Series*. Editorial Board, 2009-2016.
 - Library Quarterly*. Editorial Board, 2003-2008
 - Bulletin of the Medical Library Association*, 1994-97
 - Newsletter editor for the Chinese American Librarians Association, 1989-92
- Referee for the following journals
 - Information Processing and Management*
 - Journal of Digital Information*
 - Journal of Education for Library and Information Science*
 - Journal of Information Science*
 - Journal of Library & Information Science*
 - Journal of Library Metadata*
 - Journal of the American Society for Information Science & Technology*
 - Library and Information Science Research*
 - Library Quarterly*
- Expert reviewer, “Digital Library” course, Evaluation module, University of North Carolina, Chapel Hill, 2007-2008.
- Expert reviewer, “Information Organization” course, University of Michigan, Ann Arbor. 2007.

Catholic University of America

- Academic Senate Committee on Appointments and Promotions, 2005-2008
- Academic Senate Committee on Committees and Rules, 2009-2012, 2019-2022
- Academic Senate representative (for School of Arts & Sciences), 2017-2020

- Academic Senate representative (for School of Library & Information Science) 2003-2012
- Academic Senate Library Committee, Interim Chair (2007), Member, 2008-2012
- Academic Senate Committee on Computing, 1995-2003
- President's Administrative Council, 2010-2012
- Deans' Council, 2010-2012
- Academic Leadership Group, 2010-2012
- Graduate Board, 2010-2012
- School of Arts & Sciences, Committee on Appointments and Promotions, 2015-2019
- School of Arts & Sciences, Academic Council, 2015-2016.
- School of Arts & Sciences, Ordinary Professor Group, 2013-
- Doctoral Dissertation Defense Committee, Chair, Dept. of Psychology, 2016, 2017, 2018, 2019, 2020, 2021, 2022
- Doctoral Dissertation Defense Committee, Chair, Dept. of Education, 2014, 2015, 2017, 2018, 2019
- Doctoral Dissertation Defense Committee, Chair, National Catholic School of Social Services, 2019, 2020, 2021
- Doctoral Dissertation Defense Committee, Chair, School of Nursing, 2006, 2008
- CUA Scholarly Communication Project Team, Member (2007), Chair, 2008-2009
- Dean Search Committee, 1992-1994, 1998-1999, 2002-2003, 2006-2007
- Fulbright Review Panel, 2006
- CUA Service Learning Advisory Board, 2001-2002
- CUA Faculty Conversations on Enhancing Teaching and Learning through Technology, Planning Group, 1999-2001
- CUA Initiative on Technology and Teaching, 1998-2001

Dept. of Library and Information Science

- Admissions Committee, 2007-2009, Chair 2010-2012, Member 2013-2015, Member 2018-2022
- Appointments and Promotions Committee, 1991-2022
- Student advisement, 1990-2022
- 4+1 Bachelor to Master program, Program Advisor, 2014-2022
- LIS Minor, Advisor, 2017-2022
- Curriculum Committee, 2020-21 (Member), 2021-2022 (Chair)
- Scholarship & Awards Committee, 2016-2018, 2020-21
- Comprehensive examination editor, 2020-21
- Symposium, Colloquium, Lecture Series Committee, May 2018-May 2020 (Chair)
- Symposium and Colloquium Committee, fall 2016-May 2018
- Bridging the Spectrum Symposium session moderator, 2018, 2019, 2020, 2021

- Organization of Information Core Course Outcomes Assessment, 2019, 2020
- The Information Professions in Society Core Course Outcomes Assessment, 2019, 2020
- Community Services Librarianship, Course of Study Review and Revision, 2020
- Organization of Information, Course of Study Review and Revision, 2019
- Organization of Information, Competency and Course Offerings Review, 2018-2019
- Accreditation presentation, Chair, June 2015-August 2016
- Interim Co-Chair, June 2015-August 2016.
- Blended/OWL Learning Committee, spring 2016-2018
- Technology Committee, fall 2016-2017
- Comprehensive examination editor, 2016-2017, reader (every year since 1990)
- LIS Advisory Board, 2015-2016 (chair); fall 2016- May 2018 (member)
- Committee on Planning and Assessment, 2015-2016 (chair)
- Senior Faculty Committee, 2014-2016.
- Accreditation Steering Committee, 2014-2016 (Chair, 2015-2016)
- Accreditation Students Standard Committee, co-chair, 2014-2016
- Accreditation Mission, Goals, and Objectives Standard Committee, co-chair, 2014-2016
- Accreditation Curriculum Standard, member 2014-2-16
- Accreditation Administration and Finance Standard, member 2014-2016
- Cultural Heritage Information Management Project (IMLS-funded), Co-PI, 2012-2015
- Cultural Heritage Information Management Forum (scheduled for June 2015), Co-Organizer, 2013-2015
- Health Information Technology Interim Review Committee, 2015 (chair)
- Health Sciences Librarianship Advisory Group, 2015- (chair)
- Comprehensive examination editors, 2013-2014, 2016-2017
- National Digital Stewardship Alliance liaison, 2011-2014
- Advisory Board, Chair 2010-2012
- Academic Honesty Committee, Chair, 2008-2012
- Blended Learning Committee, 2010-2012
- Colloquium Committee, 2010-2012
- Comprehensive Examination Administration, 2010-2012
- Cultural Heritage Information Management Advisory Committee, 2010-2012 (chair), 2013-
- Curriculum Committee, 1991-2003, 2007-2009, Chair 2010-2012, member 2013-
- Curriculum Subcommittee on Comprehensive Examination, Chair 2009-2012
- Health Information Technology Advisory Board, Chair 2010-2012. Member 2013-
- Health Sciences Advisory Committee, 2009, Chair 2010-2012. Member 2013-
- HIT Expert Forum, Chair 2012. Member 2013-2015
- Health Information Technology Student Group Advisor, 2011-2012
- State Council for Higher Education of Virginia, SLIS Representative, 2010-2012
- Symposium Planning Committee, 2010-2012
- Website Management Team, Chair, 2010-2012
- Urban School Librarianship Project (IMLS-Funded), PI, 2007-2011 (chair, 2010-11)

- Failing Grades Committee, 1995-1997 (chair), 2000-2001 (chair), 2004-2005 (chair), 2007 (chair)-2011
- Faculty Search Committee, 1994-1998, 2002-2004, 2006 (chair), Fall 2007-2009, Chair fall 2009-2012
- Recruitment Committee, Chair 2010-2012
- Strategic Planning Committee, Chair 2010-2012
- Technology Committee, 2010-2012
- Accreditation Advisory Committee, 2007-2009
- Accreditation Coordinating Committee, 2007-2009
- Accreditation Steering Committee, 2007-2009
- SLIS Advisory Group, 2007-2009
- Accreditation Curriculum Standard Committee, Co-chair, 2007-2009
- Accreditation Faculty Standard Committee, Co-chair, 2007-2009
- LSC 551 Information Organization Review Team, Co-chair, 2008-2009, 2015-2016.
- Curriculum Subcommittee on Portfolios, 2009
- LSC 555 Information Systems in Libraries and Information Centers Review Team, contributor, 2008-2009
- Redesign of LSC 730 Use and Users of Libraries and Information. 2009-
- Development of a metadata institute that was taught as LSC 715 Organization of Internet Resources in 2008. The institute is being revised and will be offered in 2010 under a new course title.
- Development of lesson plans, assignments, and evaluation rubrics for LSC 606, Cataloging and Classification, for the School's NCATE accreditation. 2008
- Howard and Mathilde Rovelstad Scholarship Committee, Chair, 2004-2007
- Assistant Dean Search Committee, Chair, Fall 2007
- Liaison to the Association for Library Collections and Technical Services to bring its preconference program, Cataloging Education and Employer Expectations, to CUA during the 2007 American Library Association Annual Meeting in Washington, D.C.
- Organizer of the colloquium presentation and reception for Tamar Sadeh of Ex Libris on PRIMO June 2007
- Practicum review and design (work with potential supervisors, such as the American Indian Museum internship description revision) 2006-
- Comprehensive examinations (edits, proctoring, and grading), 1990-
- SLIS Web site redesign: Comments and suggestions. Fall 2007
- Conducted surveys of current students and alumni in preparation for the 2005 re-accreditation, 2004-2005
- Technology Committee, 1992-1999 (chair, 1996-1998), 2002-2003 (member)
- Colloquia Committee 1997-1999, 2002-2003.
- Advisor of the CUA Student Chapter of the American Society for Information Science and Technology, 2002-2003
- Visiting Professor Search Committee, 1999, 2000, 2001
- Leader, Participation in the CORC experiment, 1999-2000
- Advisor of the Special Libraries Association Student Chapter, 1993-1999; the group was recognized for outstanding leadership by SLA in 1999.

- COA planning Committee, Task Force on Electronic Presentation of SLIS Reports (team leader) 1997-1998
- COA Planning Committee, Subcommittee on Technology 1996-1998
- NLM practicum coordinator, 1997-1998
- Computer Literacy Workshops: Assisted with the development and evaluation of the workshops, 1996-1998
- Leader, Participation in the InterCat project, 1995-1997

APPENDIX B

[Library of Congress](#) >> [MARC](#) >> [Understanding MARC](#)

MARC 21 Reference Materials

[Part VII: A Summary of Commonly Used MARC 21 Fields](#)

[Part VIII: A List of Other Fields Often Seen in MARC Records](#)

[Part IX: The Leader](#)

[Part X: Field 008 for Books](#)

Part VII:

A Summary of Commonly Used MARC 21 Fields

This is a summary of the MARC 21 tags used most frequently by libraries in entering their own bibliographic records. For full listings of all MARC 21 tags, indicators, and subfield codes, see *MARC 21 Format for Bibliographic Data*.

In the explanations on these pages:

Tags -- The tags (3-digit numbers) are followed by the names of the fields they represent. In this summary, and in the *MARC 21 Format for Bibliographic Data*, if a tag can appear more than once in one bibliographic record, it is labeled repeatable (R). If it can only be used once, it is labeled non-repeatable (NR). For example, a catalog record can have several subjects, so the tags for subject added entries (6XX) are labeled repeatable (R).

Indicators -- The use of indicators is explained in fields where they are used. Indicators are one-digit numbers. Beginning with the 010 field, in every field -- following the tag -- are two character positions, one for Indicator 1 and one for Indicator 2. The indicators are not actually defined in all fields, however. And it is possible that a 2nd indicator will be used, while the 1st indicator remains undefined (or vice versa). When an indicator is undefined, the character position will be represented by the character # (for blank space).

Subfield codes -- All the data in each field (beginning with the 010 field) is divided into subfields, each of which is preceded by a delimiter-subfield code combination. The most common subfield codes used with each tag are shown. Each subfield code is preceded by the character \$, signifying a delimiter. The name of the subfield follows the code.

In general, every field **MUST** have a subfield 'a' (**\$a**). One exception that is often seen is in Field 020 (ISBN), when the ISBN information (subfield **\$a**) is unavailable but the price (subfield **\$c**) is known. Some subfields are repeatable. In this summary, repeatability is noted for only the more common repeatable subfields.

Examples: Examples follow the explanation for each field. For clarity, one space has

been placed between the tag and the first indicator, one space has been placed between the second indicator and the first delimiter- subfield code, and one space has been inserted between the delimiter-subfield code and the subfield data.

010 Library of Congress Control Number -- (LCCN)

(NR, or Not Repeatable)

Indicators undefined.

Subfield used most often:

\$a -- Library of Congress control number

Example: 010 ## \$a ###86000988#

020 International Standard Book Number -- (ISBN)

(R, or Repeatable)

Indicators undefined.

Subfields used most often:

\$a -- International Standard Book Number

\$c -- Terms of availability (often a price)

\$z -- Cancelled/invalid ISBN (R)

Example: 020 ## \$a 0877547637

040 Cataloging source -- (NR)

Indicators undefined.

Subfields used most often:

\$a -- Original cataloging agency

\$c -- Transcribing agency

\$d -- Modifying agency (R)

Example: 040 ## \$a DLC
 \$c DLC
 \$d gwhs

100 Main entry -- Personal name -- (primary author)

(NR; there can be only one main entry)

Indicator 1: Type of personal name entry element

0 -- Forename

1 -- Surname (this is the most common form)

3 -- Family name

Indicator 2 undefined.

Indicator 2 became obsolete in 1990. Older records may display 0 or 1

Subfields used most often:

\$a -- Personal name

\$b -- Numeration

\$c -- Titles and other words associated with a name (R)

\$q -- Fuller form of name

\$d -- Dates associated with a name (generally, year of birth)

Example: 100 1# \$a Gregory, Ruth W.
 \$q (Ruth Wilhelme),
 \$d 1910-

130 Main entry -- Uniform title -- (NR)

Indicator 1: Nonfiling characters

0-9 -- Number of nonfiling characters present (for initial articles, including spaces)

Indicator 2 undefined.

Indicator 2 became obsolete in 1990. (See 100 above.)

Subfields used most often:

\$a -- Uniform title

\$p -- Name of part/section of a work (R)

\$l -- Language of a work

\$s -- Version

\$f -- Date of a work

Example: 130 0# \$a Bible.
 \$p O.T.
 \$p Psalms.

240 Uniform title (NR)

Indicator 1: Uniform title printed or displayed

0 -- Not printed or displayed

1 -- Printed or displayed (most common)

Indicator 2: Nonfiling characters

0-9 -- Number of nonfiling characters present (for initial articles, including spaces)

Subfields used most often:

\$a -- Uniform title

\$l -- Language of a work

\$f -- Date of a work

Example: 240 10 \$a Ile mystérieuse.
 \$l English.
 \$f 1978

245 Title Statement (NR)

Indicator 1: Title added entry

(Should the title be indexed as a title added entry?)

0 -- No title added entry

(indicates a title main entry; i.e. no author is given)

1 -- Title added entry

(the proper indicator when an author given in 1XX; the most common situation)

Indicator 2: Nonfiling characters

0-9 -- Number of nonfiling characters present, including spaces; usually set at zero, except when the title begins with an article; e.g., for *The robe*, the second indicator would be set to 4. The letters *T*, *h*, *e*, and the space following them are then ignored in alphabetizing titles. The record will be automatically filed under "*r*" -- for *Robe*.

Subfields used most often:

\$a -- Title proper

\$h -- Medium (often used for non-book media)

\$p -- Name of part/section of a work (R)

\$b -- Reminder of title (subtitles, etc.)

\$c -- Remainder of title page transcription/Statement of responsibility

Example: 245 14 \$a The DNA story :
 \$b a documentary history of gene
 cloning /
 \$c James D. Watson, John Tooze.

246 Varying form of title (R)

Indicator 1: Note/title added entry controller

- 1 -- Note, title added entry
- 3 -- No note, title added entry

Indicator 2: Type of title

- # -- No information provided
- 0 -- Portion of title
- 1 -- Parallel title
- 4 -- Cover title
- 8 -- Spine title

Subfield used most often:

- \$a** -- Title proper

Example: 246 3# \$a Four corners power review

250 Edition statement (NR)*Indicators undefined.**Subfield used most often:*

- \$a** -- Edition statement

Example: 250 ## \$a 6th ed.

260 Publication, distribution, etc. (Imprint) (R)*Indicator 1: Sequence of publishing statements*

- # -- No information provided

*Indicator 2: Undefined**Subfields used most often:*

- \$a** -- Place of publication, distribution, etc. (R)
- \$b** -- Name of publisher, distributor, etc. (R)
- \$c** -- Date of publication, distribution, etc. (R)

Example: 260 ## \$a New York :
 \$b Chelsea House,
 \$c 1986.

300 Physical description (R)

Indicators undefined.

Subfields used most often:

\$a -- Extent (number of pages) (R)

\$b -- Other physical details (usually illustration information)

\$c -- Dimensions (cm.) (R)

\$e -- Accompanying material (for example, "teacher's guide" or "manual")

Example: 300 ## \$a 139 p. :
 \$b ill. ;
 \$c 24 cm.

440 Series statement / Added entry--Title

This field was made obsolete in 2008 to simplify the series statement. See 490 and 830.

490 Series statement (No added entry is traced from field) (R)

Indicator 1: Specifies whether series is traced (whether an 8XX tag is also present)

0 -- Series not traced

1 -- Series traced (8XX is in record)

Indicator 2 undefined.

Subfield used most often:

\$a -- Series statement (R)

\$v -- Volume number (R)

Example: 490 1# \$a Colonial American craftsmen

500 General note (R)

Indicators undefined.

Subfield used most often:

\$a -- General note (Used when no specialized note field has been defined for the information. Examples: Notes regarding the index; the source of the title; variations in title; descriptions of the nature, form, or scope of the item.)

Example: 500 ## \$a Includes index.

504 Bibliography, etc. note (R)

Indicators undefined.

Subfield used most often:

\$a -- Bibliography, etc. note

Example: 504 ## \$a Includes bibliographical references.

505 Formatted contents note (R)

Indicator 1: Type of contents note

0 -- Complete contents

1 -- Incomplete contents (used with multivolume set when some volumes are not yet published)

2 -- Partial contents

Indicator 2: Level of content designation

```
# -- Basic
```

Subfield used most often:

\$a -- Formatted contents note

Example: 505 0# \$a Pride and prejudice -- Emma
-- Northanger Abbey.

520 Summary, etc. note (R)

Indicator 1: Display constant controller

-- Summary

1 -- Review

2 -- Scope and content

3 -- Abstract

Indicator 2 undefined

Subfields used most often

\$a -- Summary, abstract, or annotation

\$b -- Expansion of summary note

<i>Example:</i>	520 ## \$a This basic guide to parliamentary procedure tells how to conduct and participate in a meeting properly.
-----------------	--

600 Subject added entry -- Personal name (R)

Indicator 1: Type of personal name entry element

- 0 -- Forename
- 1 -- Surname (this is the most common form)
- 3 -- Family name

Indicator 2: Subject heading system/thesaurus (identifies the specific list or file which was used)

- 0 -- Library of Congress Subject Headings
- 1 -- LC subject headings for children's literature
- 2 -- Medical Subject Headings
- 3 -- National Agricultural Library subject authority file
- 4 -- Source not specified
- 5 -- Canadian Subject Headings
- 6 -- Répertoire de vedettes-matière
- 7 -- Source specified in subfield \$2

(Note regarding Sears subject headings: The MARC 21 format does not provide an assigned indicator for Sears subject headings. Therefore, an indicator of 7 is used, and the MARC defined code "sears" is placed in subfield \$2.)

Subfields used most often:

- \$a** -- Personal name (surname and forename)
- \$b** -- Numeration
- \$c** -- Titles and other words associated with a name (R)
- \$q** -- Fuller form of name
- \$d** -- Dates associated with a name (generally, year of birth)
- \$t** -- Title of a work
- \$v** -- Form subdivision (R)
- \$x** -- General subdivision (R)
- \$y** -- Chronological subdivision (R)
- \$z** -- Geographic subdivision (R)
- \$2** -- Source of heading or term (used with 2nd indicator of 7)

Example: 600 10 \$a Shakespeare, William,
 \$d 1564-1616
 \$x Comedies
 \$x Stage history.

Example: 600 10 \$a Shakespeare, William,
 \$d 1564-1616
 \$x Knowledge

\$Z Rome \$v Congresses .

Notice that subfields \$v, \$x, and \$z in the 600 field are repeatable. Subfields \$v, \$x, \$y, and \$z do not have to be in alphabetical order. They will be in the order prescribed by the instructions given by the subject heading system.

610 Subject added entry -- Corporate name (R)

Indicator 1: Type of corporate name entry element

- 0 -- Inverted name (not used with AACR2)
- 1 -- Jurisdiction name
- 2 -- Name in direct order

Indicator 2: Subject heading system/thesaurus.

See indicator 2 under 600

Subfields used most often:

- \$a** -- Corporate name or jurisdiction name as entry element
- \$b** -- Subordinate unit (R)
- \$v** -- Form subdivision (R)
- \$x** -- General subdivision (R)
- \$y** -- Chronological subdivision (R)
- \$z** -- Geographic subdivision (R)
- \$2** -- Source of heading or term (used with 2nd indicator of 7)

<i>Example:</i> 610 10 \$a United States. \$b Army Air Forces \$v Biography.
--

650 Subject added entry -- Topical term (Most subject headings fit here.) (R)

Indicator 1: Level of subject

- # -- No information provided

Indicator 2: Subject heading system/thesaurus

(identifies the specific list or file which was used)

- 0 -- Library of Congress Subject Headings
- 1 -- LC subject headings for children's literature
- 2 -- Medical Subject Headings
- 3 -- National Agricultural Library subject authority file
- 4 -- Source not specified
- 5 -- Canadian Subject Headings
- 6 -- Répertoire de vedettes-matière
- 7 -- Source specified in subfield \$2

Note regarding Sears subject headings: The MARC 21 format does not provide an assigned indicator for Sears subject headings. Therefore, an indicator of 7 is used, and the MARC defined code "sears" is placed in subfield \$2.)

Subfields used most often:

- \$a -- Topical term
- \$v -- Form subdivision (R)
- \$x -- General subdivision (R)
- \$y -- Chronological subdivision (R)
- \$z -- Geographic subdivision (R)
- \$2 -- Source of heading or term used with 2nd indicator of 7)

Example: 650 #0 \$a Theater
 \$z United States
 \$v Biography
 \$v Dictionaries.

Notice that subfields \$v, \$x, and \$z in the 650 field are repeatable. Subfields \$v, \$x, \$y, and \$z do not have to be in alphabetical order. They will be in the order prescribed by the instructions given by the subject heading system.

651 Subject added entry -- Geographic name (R)

Indicator 1: undefined.

Indicator 2: Subject heading system/thesaurus.

See indicator 2 under 600

Subfields used most often:

- \$a -- Geographic name
- \$v -- Form subdivision (R)
- \$x -- General subdivision (R)
- \$y -- Chronological subdivision (R)
- \$z -- Geographic subdivision (R)
- \$2 -- Source of heading or term (used with 2nd indicator of 7)

Example: 651 #0 \$a United States
 \$x History
 \$v Chronology.

Notice that subfields \$v, \$x, and \$z in the 651 field are repeatable. Subfields \$v, \$x, \$y, and \$z do not have to be in alphabetical order. They will be in the order prescribed by the instructions given by the subject heading system.

700 Added entry -- Personal name (R)

Indicator 1: Type of personal name entry element

- 0 -- Forename
- 1 -- Surname (this is the most common form)
- 3 -- Family name

Indicator 2: Type of added entry

- # -- No information provided (most common; co-authors, editors, etc.)
- 2 -- Analytical entry (The values for Indicator 2 changed in 1994 with Format Integration, and older records may display additional values. An analytical entry involves an author/title of an item contained in a work.)

Subfields used most often:

- \$a** -- Personal name
- \$b** -- Numeration
- \$c** -- Titles and other words associated with a name (R)
- \$q** -- Fuller form of name
- \$d** -- Dates associated with a name (generally, year of birth)
- \$e** -- Relator term (such as ill.) (R)
- \$4** -- Relator code (R)

Example: 700 1# \$a Baldrige, Letitia.

710 Added entry -- Corporate name (R)

Indicator 1: Type of corporate name entry element

- 0 -- Inverted name (not used with AACR2)
- 1 -- Jurisdiction name
- 2 -- Name in direct order

Indicator 2: Type of added entry.

- See Indicator 2 under 700
- # -- No information provided
- 2 -- Analytical entry

Subfields used most often:

- \$a** -- Corporate name or jurisdiction name as entry element
- \$b** -- Subordinate unit (R)

Example: 710 2# \$a Sunburst Communications (Firm)

740 Added entry -- Uncontrolled related/analytical title (R)

Indicator 1: Nonfiling characters

0-9 -- Number of nonfiling characters present (for initial articles, including spaces)

Indicator 2: Type of added entry. See Indicator 2 under 700

-- No information provided

2 -- Analytical entry

(This field was redefined in 1994 with Format Integration. Prior to 1994, the field was also used for variant titles, such as a different wording on a spine title. In records created since Format Integration, those variant titles appear in a 246 field.)

Subfield used most often:

\$a -- Title

Example: 740 02 \$a Uncle Vanya.

800 Series added entry -- Personal name (R)

Indicator 1: Type of personal name entry element

0 -- Forename

1 -- Surname

3 -- Family name

Indicator 2 undefined.

Subfields used most often:

\$a -- Personal name

\$b -- Numeration

\$c -- Titles and other words associated with a name (R)

\$q -- Fuller form of name

\$d -- Dates associated with a name (generally, year of birth)

\$t -- Title of a work (the series)

\$v -- Volume number

Example: 800 1# \$a Fisher, Leonard Everett.
 \$t Colonial American craftsmen.

830 Series added entry -- Uniform title (R)

Indicator 1 undefined.

Indicator 2: Nonfiling characters

0-9 -- Number of nonfiling characters present (for initial articles, including spaces)

Subfield used most often:

\$a -- Uniform title

\$v -- Volume number

Example: 830 #0 \$a Railroads of America (Macmillan)

[[Back to Top of Page](#)]

Part VIII:

A List of Other Fields Often Seen in MARC Records

001	Control number
003	Control number identifier
005	Date and time of latest transaction
006	Fixed-length data elements -- additional material characteristics
007	Physical description fixed field
008	Fixed length data elements (See Part X)
022	International Standard Serial Number (ISSN)
037	Source of acquisition
041	Language code
043	Geographic area code
050	Library of Congress call number
060	National Library of Medicine call number
082	Dewey Decimal classification number (the one recommended by the Library of Congress; locally-assigned call numbers may appear elsewhere)
110	Main entry -- Corporate name (less frequent under AACR2 rules)
256	Computer file characteristics
263	Projected publication date (indicates a CIP -- Cataloging in Publication -- record)
306	Playing time
508	Creation/production credits note
510	Citation/references note (review sources)
511	Participant or performer note
521	Target audience note (first indicator: 0 = reading grade level, 1 = interest age level, 2 = interest grade level, 3 = special audience characteristics, 4 = motivation interest level)
530	Additional physical form available note
538	System details note
586	Awards note
656	Index term -- Occupation
730	Added entry -- Uniform title
852	Location

- 856 Electronic location and access
- 9XX Reserved for local use. (They are used by vendors, systems, or individual libraries to exchange additional data)

[[Back to Top of Page](#)]

Part IX:

The Leader

There are 24 positions in the Leader, numbered from 00 to 23. For fuller explanation, see the *MARC 21 Format for Bibliographic Data*.

- 00-04 Record length (calculated by the computer for each record)
- 05 Record status
 - a = increase in encoding level
 - c = corrected or revised
 - d = deleted
 - n = new
 - p = increase in encoding from prepublication (previous CIP)
- 06 Type of record
 - a = language material
 - c = printed music
 - d = manuscript music
 - e = cartographic material
 - f = manuscript cartographic material
 - g = projected medium
 - i = nonmusical sound recording
 - j = musical sound recording
 - k = 2-dimensional nonprojectable graphic
 - m = computer file
 - o = kit
 - p = mixed materials
 - r = 3-dimensional artifact or naturally occurring object
 - t = manuscript language material
- 07 Bibliographic level
 - a = monographic component part
 - b = serial component part
 - c = collection
 - d = subunit
 - i = integrating resource
 - m = monograph/item
 - s = serial
- 08 Type of control
 - # = no specified type
 - a = archival

- 09 **Character coding scheme**
 # = MARC-8
 a = UCS/Unicode
- 10 **Indicator count** (always "2")
- 11 **Subfield code count** (always "2")
- 12-16 **Base address of data** (calculated by the computer for each record)
- 17 **Encoding level**
 # = full level
 1 = full level, material not examined
 2 = less-than-full level, material not examined
 3 = abbreviated level
 4 = core level
 5 = partial (preliminary) level
 7 = minimal level
 8 = prepublication level (CIP)
 u = unknown
 z = not applicable
- 18 **Descriptive cataloging form**
 # = non-ISBD
 a = AACR2
 i = ISBD
 u = unknown
- 19 **Multipart resource record level**
 # = Not specified or not applicable
 a = Set
 b = Part with independent title
 c = Part with dependent title
- 20 **Length of the length-of-field portion** (always "4")
- 21 **Length of the starting-character-position portion** (always "5")
- 22 **Length of the implementation-defined portion** (always "0")
- 23 **Undefined** (always "0")

[\[Back to Top of Page \]](#)

Part X:

Field 008 for Books

Field 008 is used for Fixed Length Data Elements ("Fixed Field Codes"). There are 40 character positions in field 008, numbered from 00-39. Undefined positions must contain either a blank (#) or a fill character (|). Positions 00-17 and 35-39 are defined the same way for all media.

The information shown here for positions 18-34 applies only to books. For explanation of all the positions below and for positions 18-34 for other media, see the *MARC 21 Format*

for Bibliographic Data.

Note that field 008 has no indicators or subfield codes.

- 00-05 Date entered on file (YYMMDD),
where Y=year, M=month, and D=day
- 06 Type of date/publication status:
 b = no dates given; B.C. date involved
 e = detailed date
 s = single known date/probable date
 m = multiple dates
 r = reprint/reissue date (Date 1) and original date (Date 2)
 n = dates unknown
 q = questionable date
 t = publication date and copyright date
 | = no attempt to code
- 07-10 Date 1/beginning date of publication
- 11-14 Date 2/ending date of publication

Date fields contain the year(s) of publication. The type of date(s) in these elements are specified in fixed field element 06: Type of date/publication status. (For further details, see the field 008 description in the *MARC 21 Format for Bibliographic Data*.)

- 15-17 Place of publication, production, or execution
 For example:
 pk# = Pakistan
 cau = California (US)

(For a full list of codes used in these positions, see the [MARC Code List for Countries](#).)

- 18-21 Illustrations (up to 4 codes):
 # = no illustrations
 a = illustrations
 b = maps
 c = portraits
 d = charts
 e = plans
 f = plates
 g = music
 h = facsimiles
 i = coats of arms
 j = genealogical tables
 k = forms
 l = samples
 m = phonodisc, phonowire, etc.
 o = photographs
 p = illuminations
 | = no attempt to code

- 22 **Target audience:**
= unknown or not specified
a = preschool
b = primary
c = pre-adolescent
d = adolescent
e = adult
f = specialized
g = general
j = juvenile
| = no attempt to code
- 23 **Form of item:**
= none of the following
a = microfilm
b = microfiche
c = microopaque
d = large print
f = braille
r = regular print reproduction
s = electronic
| = no attempt to code
- 24-27 **Nature of contents (up to 4):**
= no specified nature of contents
a = abstracts/summaries
b = bibliographies (is one or contains one)
c = catalogs
d = dictionaries
e = encyclopedias
f = handbooks
g = legal articles
i = indexes
j = patent document
k = discographies
l = legislation
m = theses
n = surveys of literature
o = reviews
p = programmed texts
q = filmographies
r = directories
s = statistics
t = technical reports
u = standards/specifications
v = legal cases and notes
w = law reports and digests
z = treaties
| = no attempt to code

- 28 **Government publication:**
 # = not a government publication
 i = international intergovernmental
 f = federal/national
 a = autonomous or semi-autonomous component
 s = state, provincial, territorial, dependent, etc.
 m = multistate
 c = multilocal
 l = local
 z = other type of government publication
 o = government publication -- level undetermined
 u = unknown if item is government publication
 | = no attempt to code
- 29 **Conference publication:**
 0 = not a conference publication
 1 = conference publication
 | = no attempt to code
- 30 **Festschrift:**
 0 = not a festschrift
 1 = festschrift
 | = no attempt to code
- 31 **Index:**
 0 = no index
 1 = index present
 | = no attempt to code
- 32 **Undefined (since 1990)** (Earlier records may contain the values 0 or 1)
 # = Undefined
 | = no attempt to code
- 33 **Literary form:**
 0 = not fiction (not further specified)
 1 = fiction (not further specified)
 c = comic strips
 d = dramas
 e = essays
 f = novels
 h = humor, satires, etc.
 i = letters
 j = short stories
 m = mixed forms
 p = poetry
 s = speeches
 u = unknown
 | = no attempt to code
- 34 **Biography:**
 # = no biographical material
 a = autobiography
 b = individual biography

- c = collective biography
- d = contains biographical information
- | = no attempt to code

35-37 Language:

A three-letter code. For example: eng fre ger spa rus ita

(For a full list of codes used in these positions, see the [MARC Code List for Languages](#).)

38 Modified record:

- # = not modified
- x = missing characters (because of characters unavailable in MARC character set)
- s = shortened
- d = "dashed-on" information omitted
- r = completely romanized/printed cards in script
- o = completely romanized/printed cards romanized
- | = no attempt to code

39 Cataloging source:

- # = national bibliographic agency
- c = cooperative cataloging program
- d = other sources
- u = unknown
- | = no attempt to code

[[Back to Top of Page](#)]

[[Back to Table of Contents](#)] -- [[Continue to Part 11](#)]



Library of Congress

[Library of Congress Help Desk](#) (10/27/2009)

EXHIBIT 1005

DIP : Distributed Diagnosis Protocol

Luís Moura Silva and João Gabriel Silva

Laboratório de Informática e Sistemas
Universidade de Coimbra
Urb. Boavista, Lt. 1-1
3000 Coimbra - Portugal
Fax : +351.39.701266
e-mail : lams@uc.pt

Abstract

This paper presents the DIP - a Diagnosis Protocol to check the availability of processors in a distributed system. Each node maintains a Network Status Table, and failure detection is achieved by alive messages. The overhead of such messages is minimal and the main virtue of the protocol is the lower diagnosis latency when a failure is detected. Since the unavailability of a node can be the result of a processor crash or a network partition, the DIP protocol is also able to cope with partitions. The protocol can diagnose fail-stop and omission failures. Moreover, it is robust against out-of-order messages, duplicates and false-alarms.

Keywords : Distributed Diagnosis, Fault Tolerance, Membership Protocols, Network Monitoring.

1. Introduction

One of the requirements of a fault-tolerant distributed system is the failure detection capability of remote sites. Recovery and reconfiguration are other desirable properties. Since the paper of Preparata, Metze and Chien [1] there were further theoretical developments [2] in the area of distributed diagnosis. As a result, several PMC-based diagnosis algorithms have been suggested [3-9]. Recently, Bianchini *et al.* [10,11] have conducted the first implementations of system diagnosis theory in a real distributed network environment. Nevertheless, all those PMC-based algorithms have serious drawbacks:

1) when the number of node failures exceeds the connectivity of the system digraph, those algorithms simply yield false diagnostic results. 2) The PMC-model is only valid for permanent processor failures and it assumes that a node cannot fail and restart in the period between two tests. 3) Most of the algorithms require a high number of diagnostic messages and, as a result, the execution time may exceed reasonable values.

All these shortcomings make the algorithms unsuited to local area networks. Moreover, if one considers that processors may exhibit byzantine faultiness [12], PMC-based algorithms may identify

none of the faulty processors [13]. Another class of diagnosis algorithms which identifies byzantine faults has been suggested [13-15]. These algorithms are able to diagnose intermittent and malicious faults. However, the diagnosis of byzantine faults becomes extremely difficult and all the proposed solutions present a great message overhead. As practitioners, we are not willing to pay such price and we simply assume a more restricted failure model: the crash failure model. In this model it is assumed that each node can only fail in "clean" well defined ways. During its functioning the node behaves correctly and if an internal failure occurs it halts automatically. Such nodes are called fail-silent or fail-stop. A lot of algorithms [18-27] have also assumed this failure model but no one seems to perfectly suit our goals and assumptions.

2. System Model and Assumptions

The system consists of a set of N stations interconnected by a LAN with broadcast capability. Each station has access to *stable storage* which can be used to store information that must survive to failures. In perfect conditions each processor can communicate with any other processor in the network. We assume that the relation "can-communicate" between nodes is symmetric and

transitive. Two kinds of failures are considered: processor failures and communication link failures. Processor failures are assumed to be clean. When a node fails it simply stops running and does not send any arbitrary or malicious messages. Thus, the universe of failures includes stop and omission failures and no assumption is made on the number of allowable processor failures.

Network failures can be classified in hard and soft failures [16]. An hard failure is characterized by the incapability of the network to deliver any bandwidth at all. Soft failures are characterized by performance degradations or deviations from the expected behaviour. In networks, such as the Ethernet [17], which do not provide any hardware level acknowledgment, the other nodes of a network cannot distinguish an isolated node from a crashed one, since in both cases the symptom is the same - the loss of communication with the node. When an active node is not available to any other node, it means that the network was partitioned (hard failure) and both nodes are in different partitions. However, crashed nodes can recover and partitions can be reconnected. For the sake of generality let us consider the existence of two main events : crashes and restarts. Each available node is marked as UP while an unavailable node is classified as DOWN. Each processor will have its own view of the network which must be consistent with that of other nodes belonging to the same partition.

We assume that the message delivery is not reliable since the network is subjected to soft failures. Messages can be lost, delayed or delivered out-of-order. However, a network soft failure cannot produce any malicious message.

We consider that there exists a maximum message delay ∂_{max} , which reflects the expected maximum transmission time of a single message between two nodes, including the processing delay. The minimum message delay is represented by ∂_{min} . Every node should have a physical clock in order to implement timers but we do not require clocks to be synchronized.

3. Design Guidelines

The DIP protocol should be able to perform three functions :

- (1) To detect node crashes.
- (2) To notify all the network nodes of the crash of a processor (Node Down event).

- (3) and also to notify all the nodes of the restart of a crashed node (Node Up event).

Each node maintains a table with the status of the other nodes, called the *Network Status Table* (NST). Each entry of the NST has three fields : the host name, its status and the *Incarnation Number* (IN). The IN number is helpful to distinguish duplicates and out-of-order messages, as will be explained in §4.4.

Since failures are the exception rather than the rule, the first imperative is to reduce the overhead during normal operation. The protocol must assure a minimal number of messages during steady-state operation. Another important goal is to reduce the diagnosis latency to a minimal value, i.e., the protocol must provide timely failure detection. This is a paramount figure of merit to allow an effective recovery/reconfiguration of distributed applications. When a node failure occurs all the other nodes must be acquainted with that as quickly as possible. As failures and restarts are asynchronous events, the protocol should be robust against their occurrence during its execution. Other important issue is the handling of network partitions. The other desired goals are efficiency, simplicity and correctness of the protocol.

4. Design Details

4.1 The Detection of Failures

As a fail-silent processor only stops when an internal fault occurs the failure can be simply detected by background *I_AM_ALIVE* messages. The network nodes are organized on a virtual ring and all the sites are numbered from 0 up to N-1, along the path of the virtual ring. It should be noted that the ring does not reflect the physical organization of the network. The order of the nodes in the ring follows the order of the Status Table, which is unique to all the processors in the network.

Each processor P_i sends periodically an *I_AM_ALIVE* message to the next neighbour $P_{((i+1) \bmod N)}$ while it is checking for the messages sent by the previous processor $P_{((i-1) \bmod N)}$. Each processor has two independent processes to perform that job.

- 1) The *Heartbeat* process is periodically awakened by a timer interrupt in order to send a message to the next neighbour. This timer is represented by T_H and is set to α .

2) The detection of failures is done by the Watchdog process, which is listening to the messages sent by the previous neighbour. Every time it receives an `I_AM_ALIVE` message it triggers a timer T_W with $((\Omega+1) \cdot \alpha + \partial_{\max} - \partial_{\min})$, where Ω is the number of omission failures or lost messages that can be tolerated. If the timer expires, this means that the previous node has crashed and the Node Down Protocol is started for that node.

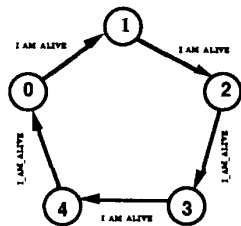


Figure 1 : Virtual Ring

4.2 Node Down Protocol

In order to reduce the diagnosis latency, the node should notify all the others immediately by a broadcast message to the network, announcing the `Node_Down` event of its neighbour.

Take as an example the network of figure 2. One can notice that node #2 has crashed and its tester (node #3) will detect that when its timer T_W expires. The status of node #2 is marked as `DOWN` in its Network Status Table, and it will send a broadcast message to the network, which includes the number of the node and the `NST(2).IN : NODE_DOWN(2, 1)`.

In a meantime, a timer T_{ack} will be set to $2 \cdot \partial_{\max}$. The node will inspect its status table to find the available nodes and waits for the acknowledge of every UP node. On the other hand, when a node receives a `NODE_DOWN` broadcast it replies with a `ND_Ack` message to node #3. It should be noted that each acknowledge is also a broadcast message. This may be useful if a particular node does not receive the broadcast message of #3 it may be informed of that by one of the acknowledge messages. When node #3 receives all acknowledges this phase is finished and the protocol returns to the steady state. However, if the timer T_{ack} expires and some reply is missing, the message `NODE_DOWN(2, 1)` is retransmitted again (until a maximum number of retries is reached).

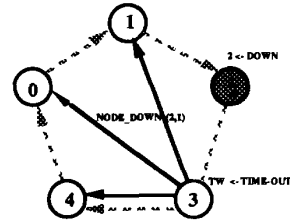


Figure 2 : Detection of a Failure

It may happen that the unavailability of node #2 is not the crash of a single processor, but instead the network may be partitioned. In this case, some nodes may not answer to the message and should also be classified as `DOWN`. In fact, if after the retransmission, node #3 does not receive any reply then it will start again this protocol announcing the unavailability of those nodes which remained in the other partition of the network.

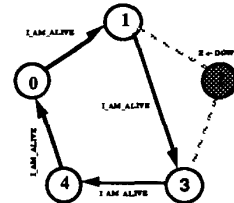


Figure 3 : Reconfiguration

After the notification phase of a node crash, the testing structure has to be adapted to a new configuration. Each node that had been notified that its tester neighbour has crashed must identify the next available node, by searching the Status Table (see example of node #1 in figure 3). The detector node (#3) will also have a new tested node (#1).

4.3 Node Up Protocol

When a node rejoins the network it must tell the others that it is back in operation and must also determine which nodes are UP and reachable. Therefore, when a node recovers it reads the NST from stable storage to main memory. The status of all the other nodes are set to `DOWN`. Then it increments its `IN` value, which is also stored in stable storage, and sets to zero all the counters of the other nodes. Finally, it broadcasts a `NODE_UP` message to notify all the active nodes in the

network. A timer T_{up} is activated. This message must be acknowledged by the other available nodes. The replies will be used by the restarted node to achieve its new network view. Furthermore, to guarantee a distributed consistency, node #2 must verify whether all the nodes it can reach share the same NST.

Let us consider the previous scenario : node #2 is DOWN and its IN is equal to 1. Now, suppose that it recovers and rejoins the network. First of all, its IN counter will be incremented, and it will send a NODE_UP broadcast message to the network, which includes the new IN (see figure 4).

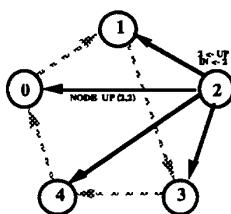


Figure 4 : Node #2 Rejoins the Network

When each of the other nodes receive the NODE_UP(2,2) message they check the 2nd entry of its NST, and the status of node #2 is marked as UP. Then, they send a NU_Ack message to node #2, which contains their own INs. Each node also appends a list of its DOWN nodes into that message, including the corresponding INs. When the timer T_{up} expires, node #2 will have a complete information about the network status. Likewise, to assure a distributed agreement, node #2 must check the information it had received from the NU_Ack messages. The DOWN lists which were piggybacked into the acknowledges must be all consistent. Otherwise, node #2 will have to broadcast the result of the comparison to force a distributed consensus.

4.4 Protocol Integration

As the two sub-protocols may execute simultaneously they must interact with each other. Let us consider the following situation : node #2 has crashed ; node #3 detects its failure and sends a NODE_DOWN broadcast message to the network. However node #2 is DOWN for only a short time, after that it comes up again and rejoins the network by sending a NODE_UP message. Yet, an

inconsistent scenario may occur: Node #0 receives the sequence (NODE_DOWN, NODE_UP) while node #1 receives the (NODE_UP, NODE_DOWN) sequence. Fortunately, the *Incarnation Number* can solve this inconsistency. Every message is tagged with the IN and each receptive node will compare the incoming IN with the value stored in its NST. When a host receives a NODE_UP message from node #K and the incoming IN exceeds NST(K).IN, the table is automatically updated. If the incoming IN is similar it is recognized as a duplicate message and no action is performed. In either cases a reply (NU_Ack) is sent. When a node receives a NODE_DOWN message from node #X notifying the crash of its neighbour (node #Y) it checks in its table the status and the current IN of node #Y. If #Y is UP and the incoming IN exceeds the NST(Y).IN, then #Y is marked as DOWN and the local IN is updated. In normal situations, the incoming IN is similar to the local value. In this case, only the status of node #Y is updated. On the other hand, if #Y is already DOWN the NST(Y).IN will only be updated if the incoming IN exceeds the local IN. In either case the receiving node will reply to each message with its own IN.

Taking the previous example, let us assume that node #2 was running with IN=1. By the time it had crashed, node #3 would have sent a NODE_DOWN message announcing this event (with IN=1). When the node recovers, it would increment IN to 2.

Node #0 would receive the messages in the correct order. However, when node #1 received the message NODE_UP(2,2) it would realize that node #2 had crashed and came up again without being aware of that. It would set NST(2).IN to 2 and send a NU_Ack message to node #2, which would include its own IN (NST(1).IN).

After having received the NODE_DOWN(2,1) message it would conclude that the message was out-of-date since the incoming IN was less than the stored one. In spite of that, a reply message (ND_Ack) would be sent to node #3.

4.5 False Alarms

A *false alarm* occurs when an active node receives a NODE_DOWN broadcast message notifying its own failure. It may happen if the number of lost messages exceeds the tolerated level. A network soft failure can be the cause of that. In this case, the node will react to that "injury" by broadcasting a

I'M_NOT_DOWN message to the network with its current IN. When the other nodes receive such message they undo the action taken by the reception of the *false alarm* message. This message is only taken into account if the incoming IN is equal to the stored value.

4.6 Coping with Network Partitions

Network partitioning is an event which causes the network to divide into two or more sub-networks. The worst case is when a single node is isolated from the rest of the network due to a crash of its communication processor. Partitioning should be detected during the execution of the Node Down Protocol.

To simplify the explanation let us consider the network represented in figure 5. Due to a communication link breakdown or to some other network component (e.g. gateway, bridge, router...) failure, the network was split into two sub-networks. The detection of this partitioning begins with the detection of two single node failures. Node #2 and node #0 will detect the crashes of node #1 and node #4, respectively. Each of the detectors will send a NODE_DOWN broadcast and then wait for the acknowledge messages. As Node #2 will note the lack of response from node #0 it will repeat the Down protocol for node #0. On the other hand, this node only receives the ND_Ack from node #1. Thus, it will broadcast to its sub-network the failure of node #2 and #3. Only node #1 will hear these messages.

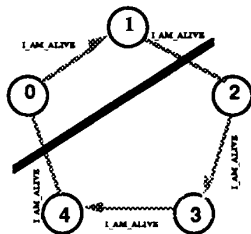


Figure 5 : Network Partitioning

The final result of this phase is represented in figure 6. Two partitions have been identified (P1 and P2) and the main graph of test has been divided into two independent graphs. To the nodes of partition P1, the nodes belonging to P2 are marked as DOWN, and vice-versa. The protocol guarantees that each time a network partitioning is detected this is assumed at

both sides of the "wall" between partitions.

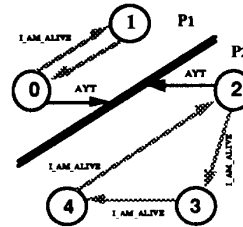


Figure 6 : Partitioning Reconfiguration

Furthermore, it is necessary to assure the detection of the possible merge of these partitions. This job is assigned to both *detector* nodes (#0 and #2). Each one will resume a *dummy* process that periodically sends a AYT² broadcast message to the network, which contains the identification of the sender and its IN. A rule is established to the nodes of the same partition : they do not answer to those messages. For instance, in partition P1 node #0 sends periodically an AYT broadcast message. Since node #0 is UP for node #1, it will not reply.

Those messages may pass through the partition "wall", and can be heard by the other nodes only when the two sub-networks get merged. Each node that receives an AYT message from a sender judged as DOWN, will acknowledge it with a YIA³ message. This reply contains the node identification, its current IN and is also broadcasted. A YIA works like a NODE_UP message with just one exception : if the incoming IN is similar to or exceeds the local IN, the message is accepted and the table is updated for the sender of the message. If the incoming IN is smaller, it is classified as obsolete. In both cases, a reply is sent to the network.

In the previous example, nodes #2 and #0 kept watching the other partition. They periodically send a AYT messages to the network. While the partition is maintained, those messages are not acknowledged. When the clusters get reconnected, node #1 and #0 hear the AYT(2) while nodes #2, #3 and #4 hear the AYT(0) message. They answer to those messages with a broadcast YIA and a special Node_Up protocol is started in each node. After that phase, the network recovers to the original configuration.

² AYT means "Are You There?"

³ YIA means "Yes, I Am !"

As it is not possible to distinguish a node failure from a network partition, every time a node detects a node crash it resumes that *dummy* process, which is suspended when all the failed nodes it has detected get up again. As it is represented in figure 7 it may happen that network partitioning does not affect a contiguous set of nodes (as in fig.6), but instead an alternate set along the path of the virtual ring. This is the worst case in terms of the number of messages, as will be seen in the next section.

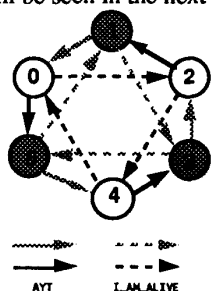


Figure 7 : Special Partitioning

5. Analysis of the Protocol

The original protocol uses 8 different messages: *I_AM_ALIVE*, *NODE_UP*, *NU_Ack*, *NODE_DOWN*, *ND_Ack*, *I'M_NOT_DOWN*, *AYT*, *YIA*.

In steady-state operation, the protocol requires the minimum number of diagnostic messages. In a network of N active processors, N messages are sent to the network between each testing interval. When a node failure occurs, it is detected by its neighbour in a finite interval called *failure detection latency*. This latency depends on how much time has elapsed between the time the last *I_AM_ALIVE* message was sent and the failure. In the worst case, the failure occurs immediately after the last message is sent out, while in the best case the node fails just before sending the message. So, the detection latency falls between (∂_{max}) and the period of the Watchdog Timer: $((\Omega+1).\alpha + \partial_{max} - \partial_{min})$.

Let us now refer to the time that it takes to notify the failure to all the other nodes the *diagnosis latency*. In the DIP approach, the path length of the diagnosis information is equal to 1 and the maximum diagnosis latency is just ∂_{max} . The diagnosability of the protocol is equal to $(N-1)$, and the number of messages necessary to perform the *Node_Down* protocol is bound to $(N-1)$ (1 broadcast

and $(N-2)$ replies)⁴. To notify a *Node_Up* event it only requires N messages. However, if we have a network with (S) subnets connected through routers, the protocol has to send (S) *NODE_UP* messages, one for each subnet, and the diagnosis latency is also increased by the processing time of each intermediate router. The message count will be bound to $(S+N-2)$ for the *Node Down* Protocol, while the *Node Up* protocol will use $(S+N-1)$ messages.

Now, let us consider the partitioning of the network into P partitions. If the set of nodes of each partition is contiguous as in figure 6 the number of messages will be $(P+N) : (P)$ *AYT* messages and (N) *I_AM_ALIVE* messages. In the worst case, as shown in figure 7, the message count will be bound to $(P.N)$. When only one node has become isolated, the number of messages will be $(N+1)$.

6. Relation to Previous Work

In this section, we will briefly discuss others algorithms found in the literature, and a comparison with the DIP will be made. The implementation of the *NEW-SELF* algorithm, conducted by Bianchini *et al.* [10] has shown that it requires a high number of diagnostic messages. It was concluded that such algorithm is not feasible to real systems. In the same paper, a new algorithm is proposed: the *EVENT-SELF*. This one reduces the message count during normal operation but if a failure occurs it produces a packet count higher than the one of *NEW-SELF*. Moreover, an experiment with 20 processors has shown that the complete diagnosis of one processor failure would take about 154 seconds. Both algorithms have a static testing graph, which means that the number of faulty units is bounded to a well established limit (t). If the number of processor failures exceeds that number (t), the network becomes undiagnosable.

In order to solve these serious problems, Bianchini proposed a new algorithm : the *Adaptive DSD* [11]. This algorithm can diagnose any number of failures since it has an adaptive testing graph. This represents some improvements, but it also has some shortcomings. Since it is a *PMC*-based algorithm it does not support that a node fails and recovers between two tests executed by another node. The algorithm is not able to diagnose link failures, and the path length of the diagnosis information is bound

⁴ Assuming no retransmissions.

to $\log_2 N$. An experiment in a network of 60 processors with a version that uses an asymmetric forwarding scheme has shown that the overall diagnosis latency was about 90 seconds for one processor failure. The version that assumes the original model of forwarding of the PMC-model has an overall diagnosis latency of 450 seconds.

If one wants an efficient algorithm the latency should be drastically reduced and the number of messages minimized. The algorithms that assume the crash failure model are the most suitable and can be easily applied in real systems since they require a lower message count and are much more simple.

A centralized approach was chosen in the Publishing system [18]. In this case, a dedicated node, called the recorder, has a watchdog process for each processor of the network. If no messages are received in a time limit, the processor is considered to have crashed. This approach is simple but has a weak point in the recording node. Another centralized approach was chosen in [19], which uses an election algorithm to choose the master node.

At the other extreme, we found the detection mechanism of the TANDEM NonStop system [20], where each node tests every other node in the network. The DIP approach falls between the centralized and the fully distributed approach.

A practical algorithm can be found in [21]. It has some similarities with the DIP. However, that proposal has some shortcomings. First, it leaves the failure detection job to the application. Second, when a node fails, the number of diagnosis messages is $2N^2 - 6N + 4$ (where N is the number of nodes). This means that the network will be flooded with messages, and the diagnosis latency will suffer from that.

The algorithm presented by Walter [22] is similar to the DIP - it also organizes the network as a virtual ring of processors. But, one may notice that it is not robust against lost messages, duplicates, out-of-order messages and false alarms, since it assumes a reliable network. The critical message path of the diagnosis is 3, while the critical path of the DIP is just 1, which means that the diagnosis latency is at least 3 times higher than the latency of DIP. During steady-state operation the RSC protocol always requires the same number of messages: $P \cdot N$.

Another algorithm implemented in the SDD/1's RelNet [23] provides a systematic detection of failures and node recoveries. Nevertheless, that

algorithm is not robust against network partitions. Another set of algorithms has been proposed for synchronous distributed systems [24-27].

7. Implementation Features

The first release of the DIP was implemented on a set of SUN workstations which communicate through an Ethernet local area network. It was written in C using sockets UDP/IP. The protocol itself was running as a background process. The initialization module stores in stable storage the list of the processors in the network. This list is sorted by the Internet host address. As this number is unique to each processor it assures that all the hosts generate identical lists. All the crash and recovery events are stored in the *Crash Log* file, which can be inspected for failure analysis. A query interface is also available to users asking for the network status. The parameters chosen were the following: $\alpha = 1$ second and $\Omega = 2$.

The DIP provides a failure detection capability to be incorporated in a recovery mechanism, based on a distributed checkpointing algorithm presented elsewhere [28].

8. Conclusions and Future Work

We have presented a protocol for checking the availability of sites in local area networks. During steady-state operation it uses the minimum number of messages, while in case of failure it assures the lowest diagnosis latency. Its testing graph is adaptive which provides a complete diagnosability. It is also robust against false alarms, out-of-order messages and network partitions. The only shortcoming is its reduced failure model. If anyone wants to extend the model should try one of the techniques presented by Neiger and Toueg [29] to automatically translate this algorithm, that only tolerates crash failures, into another one that tolerates more "severe" failures (i.e. arbitrary or byzantine). As future work we are planning to implement the protocol on a network of transputers.

Acknowledgments

L.M.Silva was supported by JNICT under the Programa Ciência (BM-38-90-IA). This work was partially supported by INIC and by JNICT under contract PMCT/C/TIT/451/90.

References

1. F.P.Preparata,G.Metze and R.T.Chien . "On the connection assignment problem of diagnosable systems" , IEEE Transactions on Electronic Computing, 16, Dec.1967, pp.848-854
2. S.E.Kreutzer and S.L.Hakimi. "System - Level Fault Diagnosis : A Survey", Euromicro Journal , May 1987, pp.323-330
3. J.G.Kuhl and S.M.Reddy. "Distributed Fault-Tolerance for Large Multi-Processor Systems", Proc. 7th Annual Symposium on Computer Architectures, May 1980,pp. 23-30
4. J.G.Kuhl and S.M.Reddy. "Fault-Diagnosis in Fully Distributed Systems", Proc. FTCS-11, 1981 pp. 100-105
5. S.H.Hosseini, J.G.Kuhl and S.M.Reddy. "On Self-Diagnosis of the Distributed Systems", Proc. FTCS-15, 1985, pp. 30-35
6. S.H.Hosseini, J.G.Kuhl and S.M.Reddy. "A Diagnosis Algorithm for Distributed Computing Systems with Dynamic Failure and Repair", IEEE Trans.on Computers,March 1984,pp.223-233
7. S.E.Kreutzer and S.L.Hakimi. "Distributed Diagnosis and the System User", IEEE Trans. on Computers, January 1988, pp. 71-78
8. N.H. Vaidja and D.K. Pradhan. "System Level Diagnosis : Combining Detection and Location", Proc. FTCS-21, 1991 , pp.488-495
9. A.Baghi and S.L.Hakimi. "An Optimal Algorithm for Distributed System Level Diagnosis", Proc. FTCS-21 , 1991, pp. 214-221
10. R.Bianchini , K.Goodwin and D.S.Nydic. "Pratical Application and Implementation of Distributed System-Level Diagnosis Theory " Proc. FTCS-20 , 1990 , pp. 332-339
11. R.Bianchini and R.Buskens. " An Adaptive Distributed System-Level Diagnosis Algorithm and Its Implementation", Proc. FTCS-21, 1991, pp. 222-229
12. L.Lamport , R.Shostak and M.Pease. "The Byzantine Generals Problem", ACM Trans. on Programming Languages and Systems, V. 4 1982, pp.382-401
13. K.V.S.Ramarao and J.C.Adams. "On the Diagnosis of Byzantine Faults", Proc. of the 7th Symposium on Reliable Distributed Systems , 1988, pp.144-153
14. K.C.Shin and P.Ramanathan. "Diagnosis of Processors with Byzantine Faults in a Distributed Computing System", Proc. FTCS-17, 1987 , pp.55-60
15. J.C.Adams and K.V.S. Ramarao. "Distributed Diagnosis of Byzantine Processors and Links", Proc. 9th Int. Conf. on Distributed Computing Systems, 1989 , pp.562-569
16. R.A. Maxion and F.E. Feather . "A Case Study of Ethernet Anomalies in a Distributed Computing Environment ", IEEE Transactions on Reliability , Vol 39, October 1990 , pp.433-443
17. J.F. Schock et al. " Evolution of the Ethernet Local Computer Network", IEEE Computer , August 1982 , Vol. 5 (8) , pp. 10-27
18. M.L.Powell and D.L.Presoto. "Publishing : A Reliable BroadcastCommunication Mechanism", Proc. 9th Symposium on Operating Systems Principles, 1983 , pp.100-109
19. T. Becker. "Keeping Processes Under Surveillance ", Proc. 10th Symposium on Reliable Distributed Systems,1991,pp.198-205
20. C.I. Dimmer. "The Tandem Non-Stop System" , in *Resilient Computing Systems* , Ed. by T. Anderson,1985, pp. 178-196
21. S.Bruso. "A Failure Detection and Notification Protocol for Distributed Computing Systems", Proc. 5th Int. Conf. on Distributed Computing Systems, 1985, pp.116-123
22. B.Walter. "A Robust and Efficient Protocol for Checking the Availability of Remote Sites ", Computer Networks, Vol. 6, 1982 ,pp. 173-188
23. M.Hammer and D.Shipman . "Reliability Mechanisms for SDD-1 : A System for Distributed Databases", ACM Transactions on Database Systems, Vol.5 (4) , Dec. 1980, pp. 431-466
24. F.Cristian."Agreeing on Who Is Present and Who Is Absent In a Synchronous Distributed System" Proc. FTCS-18, 1988, pp. 206-211
25. H.Kopetz,G.Grunsteidl and J.Reisinger. "Fault-Tolerant Membership Service in a Synchronous Distributed Real-Time System ", in *Dependable Computing for Critical Applications*, Springer Verlag, 1991
26. B.Walter. "Network Partitioning and Symmetric Surveillance Protocols", Proc. 5th Int. Conf. on Distributed Computing Systems,85,pp.124 -129
27. A.V.Ma and G.G.Belford."A Failure and Recovery Detection Protocol for Optimistic Partitioned Operation in Distributed Database Systems", Proc. 6th Int. Conf. on Distributed Computing Systems,1986, pp.540-547
28. L.M.Silva , J.G. Silva. "Global Checkpointing for Distributed Programs", Proc. 11th Symp. on Reliable Distributed Systems, pp. 155-162, 1992, Houston (USA)
29. G.Neiger and S.Toueg . " Automatically Increasing the Fault-Tolerance of Distributed Systems", Proc. 7th Symp. on Principles of Distributed Computing, 1988, pp.248-262

APPENDIX 1005A

LIBRARY OF CONGRESS



0 003 462 065 9

MICRO-
PROCESSING
AND
MICRO-
PROGRAMMING

38-39
SEPT.-DEC.
1993

QA 76
5
.E89A

Microprocessing and Microprogramming

VOLUME 38, NUMBERS 1-5
SEPTEMBER 1993

ISSN 0165-6074
MMICDT 38 (1-5) (1993) 1-862

The EUROMICRO Journal

**COMPLETE VOLUME:
PROCEEDINGS EUROMICRO 93
OPEN SYSTEM DESIGN:
HARDWARE, SOFTWARE AND
APPLICATIONS
Barcelona, September 6-9, 1993**

NORTH-HOLLAND

Published for EUROMICRO by NORTH-HOLLAND
(a division of Elsevier Science Publishers B.V.)

Ex.1007

MICROPROCESSING AND MICROPROGRAMMING
VOLUME 38, NUMBERS 1-5, SEPTEMBER 1993

CONTENTS

The contents of this volume are on pp. xv-xxiii



0165-6074(199309)38:1/5;1-1



Microprocessing and Microprogramming

The Euromicro Journal

Editors-in-chief

Mariagiovanna Sami
Politecnico Istituto di Elettronica
Piazza L. da Vinci, 32
I-20133 Milano, Italy
Tel.: (2) 2399 3516; telefax: (2) 2399 3411;
email: sami@ipmef1.polimi.it

Lutz Richter
Univ. Zürich-Irchel
Institut für Informatik
Winterthurerstrasse 190
CH-8057 Zürich, Switzerland
Tel.: (1) 257 4330; telefax: (1) 363 0035;
email: richter@ifi.unizh.ch

Editorial board

Software engineering
Gerhard Chroust
Johannes Kepler Universität Linz
Institut für Systemwissenschaften
Abt. für Systemtechnik und Automation
Altenbergerstraße 69
A-4040 Linz, Austria
Tel.: +43 (732) 2468-866
Telefax: +43 (732) 2468-822

Software and system architectures

Peter Milligan
Queen's University of Belfast
Department of Computer Science
University Road
Belfast, BT7 1NN,
United Kingdom
Tel.: +44 (232) 245 133 ext. 3293/3229
Telefax: +44 (232) 331 232
email: p.milligan@qub.ac.uk

Hardware architectures, communications and networks, embedded and real-time systems

N. Scarabottolo
Politecnico di Milano
Istituto di Elettronica
Piazza L. da Vinci, 32
I-20133 Milano, Italy
Tel.: +39 (2) 2399 3512
Telefax: +39 (2) 2399 3411

Distributed systems

Ferenc Vajda
KFKI Research Inst. for Measurement and
Computing Technique
Dept. MSZKI, P.O. Box 49
1525 Budapest 114, Hungary
Tel.: +36 (1) 169 5532; telefax: +36 (1) 169 5532
email: vajda@iif.kfki.hu

Parallel architectures

Stephen Winter
Polytechnic of Central London
115 New Cavendish Street
London, W1M 8JS,
United Kingdom
Tel.: +44 (71) 911 5099
Telefax: +44 (71) 911 5089

A list of EUROMICRO Directors and Correspondents is given in the back of this issue.

© 1993, Elsevier Science Publishers B.V. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publisher, Elsevier Science Publishers B.V. Copyright and Permissions Department, P.O. Box 521, 1000 AM Amsterdam, Netherlands.

Submission of an article for publication implies the transfer of the copyright from the author(s) to the publisher and entails the author's irrevocable and exclusive authorization of the publisher to collect any sums or considerations for copying or reproduction payable by third parties (as mentioned in article 17 paragraph 2 of the Dutch Copyright Act of 1912 and in the Royal Decree of June 20, 1974 (S. 351) pursuant to article 16b of the Dutch Copyright Act of 1912) and/or to act in or out of Court in connection therewith.

Special regulations for readers in the USA—This journal has been registered with the Copyright Clearance Center, Inc. Consent is given for copying of articles for personal or internal use, or for the personal use of specific clients. This consent is given on the condition that the copier pays through the Center the per copy fee stated in the code on the first page of each article for copying beyond that permitted by Sections 107 or 108 of the US Copyright Law. The appropriate fee should be forwarded with a copy of the first page of the article to the Copyright Clearance Center, Inc., 27 Congress Street, Salem, MA 01970, USA. If no

code appears in an article, the author has not given broad consent to copy and permission to copy must be obtained directly from the author. All articles published prior to 1981 may be copied for a per-copy fee of US \$ 2.25, also payable through the Center. This consent does not extend to other kinds of copying such as for general distribution, resale, advertising and promotion purposes, or for creating new collective works. Special written permission must be obtained from the publisher for such copying.

Special regulations for authors—Upon acceptance of an article by the journal, the author(s) will be asked to transfer copyright of the article to the publisher. This transfer will ensure the widest possible dissemination of information under the US Copyright Law.

This journal is printed on acid-free paper.
Published 15 times a year Printed in The Netherlands

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

Although all advertising material is expected to conform to ethical standards, inclusion in this publication does not constitute a guarantee or endorsement of the quality or value of such product or of the claims made of it by its manufacturer.



EUROMICRO 93

Nineteenth EUROMICRO conference

Barcelona

September 6-9, 1993

Open System Design:
Hardware, Software and Applications

Programme

Welcome to Barcelona

The nineteenth EUROMICRO Conference will be held from September 6th to 9th at the Polytechnic University of Catalunya in Barcelona.

The Conference main theme of this year is "Open System Design: Hardware, Software and Application". The purpose of the Conference is to bring together people from Business, Industry, Research, Government and Academia who are interested in all problems related to microelectronics and computing. EUROMICRO 93 aims to cater for a "state of the art" coverage of current developments and future trends.

The Polytechnic University of Catalunya and the organizers extend a warm welcome to the EUROMICRO 93 Conference and its participants.

Programme

The EUROMICRO Conference is a major annual event around which a collection of related events congregate. EUROMICRO 93 will include a mainstream programme based on Keynote Sessions, Scientific Paper Sessions organized in four parallel tracks and Short Note sessions. Several other activities are organized such as a Tutorial and several Social Events.

Euromicro 93 has attracted a large number of submitted papers which were carefully reviewed - four referees each - by an international panel of 166 referees. The Programme Committee selected 94 papers arranged in 33 sessions.

Keynotes

Each day of the conference an outstanding keynote speaker will spotlight the theme of the Conference from a different point of view. Keynote addresses will be given by:

M. Silva of University of Zaragoza (E) on "Formal modelling and analysis of architectures using Petri Nets"

L.A. Zadeh of University of California (USA) on "Fuzzy Logic, Neural Networks and Soft Computing"

J.A.B. Fortes of Purdue University (USA) on "Matching algorithms and architectures"

Scientific and Technical Programme

The Scientific and Technical Programme has been organized by Programme Subcommittees.

- Software Engineering
Coordinator: P. Milligan (UK)
- System Design
Coordinator: A. Pawlak (D)
- Computer Communications and Networks

- Coordinator: K. Tarnay (H)
- Parallel Processing
Coordinator: S. Winter (UK)
- Components and VLSI Design
Coordinator: A. Antola (I)

Short Notes

In parallel there is a Programme of Short Note sessions with contributions providing topical reports of recent works in all fields relevant to the Conference. The detailed Short Note Programme will be available at the Conference.

Tutorial

Monday, September 6th, features the tutorial Applying Concurrent Engineering in a Software Process by Marko Heikkinen (SF).

Social programme

During the EUROMICRO 93 Conference the following events are scheduled:

On Monday, September 6th, a Get-Together Party will be offered on the University campus (building TG on the map) from 17:00 - 18:30.

On Tuesday, September 7th, a reception by the Mayor of Barcelona will be held at 19:00 at the "Palau de Pedralbes", a palace located near the campus. A classical music quartet will give a performance and during the reception a cocktail will be offered.

On Wednesday, September 8th, at 21:00, a Banquet will be organized in the "Poble Espanyol", a charming place near the Olympic area, where you can visit some samples of architectural works from different places in Spain.

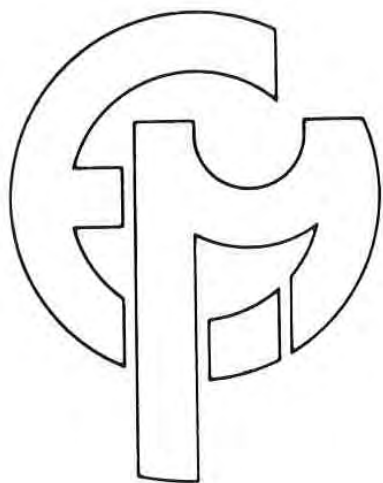


La Sagrada Família

Microprocessing and Microprogramming

The Euromicro Journal

Volume 38 (1993)



NORTH-HOLLAND



Microprocessing and Microprogramming

The Euromicro Journal

Editors-in-chief

Mariagiovanna Sami
Politecnico Istituto di Elettronica
Piazza L. da Vinci, 32
I-20133 Milano, Italy
Tel.: (2) 2399 3516; telefax: (2) 2399 3411;
email: sami@ipmel1.polimi.it

Lutz Richter
Univ. Zürich-Irchel
Institut für Informatik
Winterthurerstrasse 190
CH-8057 Zürich, Switzerland
Tel.: (1) 257 4330; telefax: (1) 363 0035;
email: richter@ifi.unizh.ch

Editorial board

Software engineering

Gerhard Chroust
Johannes Kepler Universität Linz
Institut für Systemwissenschaften
Abt. für Systemtechnik und Automation
Altenbergerstraße 69
A-4040 Linz, Austria
Tel.: +43 (732) 2468-866
Telefax: +43 (732) 2468-822

Software and system architectures

Peter Milligan
Queen's University of Belfast
Department of Computer Science
University Road
Belfast, BT7 1NN,
United Kingdom
Tel.: +44 (232) 245 133 ext. 3293/3229
Telefax: +44 (232) 331 232
email: p.milligan@qub.ac.uk

Hardware architectures, communications and networks, embedded and real-time systems

N. Scarabottolo
Politecnico di Milano
Istituto di Elettronica
Piazza L. da Vinci, 32
I-20133 Milano, Italy
Tel.: +39 (2) 2399 3512
Telefax: +39 (2) 2399 3411

Distributed systems

Ferenc Vajda
KFKI Research Inst. for Measurement and
Computing Technique
Dept. MSZKI, P.O. Box 49
1525 Budapest 114, Hungary
Tel.: +36 (1) 169 5532; telefax: +36 (1) 169 5532
email: vajda@iif.kfki.hu

Parallel architectures

Stephen Winter
Polytechnic of Central London
115 New Cavendish Street
London, W1M 8JS,
United Kingdom
Tel.: +44 (71) 911 5099
Telefax: +44 (71) 911 5089

A list of EUROMICRO Directors and Correspondents is given in the back of this issue.

© 1993, EUROMICRO Association for Microprocessing and Microprogramming

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owner.

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein. Although all advertising material is expected to conform to ethical standards, inclusion in this publication does not constitute a guarantee or endorsement of the quality or value of such product or of the claims made of it by its manufacturer.

Abstracted/Indexed in: ABI Inform, ACM Computing Reviews, AI Abstracts, CAD/CAM Abstracts, Compumath Citation Index, Computer Abstracts, Computer Literature Index, EIC Intelligence, Engineering Index/Compendex, INSPEC Information Services, Research Alert

Published 15 times a year
Printed on acid-free paper

Printed in The Netherlands

OPEN SYSTEM DESIGN: HARDWARE, SOFTWARE AND APPLICATIONS

Nineteenth EUROMICRO Symposium on
Microprocessing and Microprogramming
(EUROMICRO 93)

Barcelona, September 6–9, 1993

edited by
Ferenc Vajda
and
Benny Graff Mortensen



1993

NORTH-HOLLAND — AMSTERDAM • LONDON • NEW YORK • TOKYO

Table of Contents

Chairmen's Introduction	vii
Program Chairman's Introduction	ix
Reviewers	xi
Session Chairpersons	xiii
K1: Petri Nets applied to the Modelling and Analysis of Computer Architecture Problems (Invited paper) M. Silva and J.M. Colom (E)	1
K2: Fuzzy Logic, Neural Networks and Soft Computing (Invited paper) L.F. Zadeh (USA) (<i>abstract</i>)	13
K3: Matching Algorithms and Architectures (Invited paper) J.A.B. Fortes (USA) (<i>abstract</i>)	15
A1: Interconnection Networks Chairperson: K. Tarnay (H)	17
Hardware Features of the Static Communication Network of a Parallel Architecture V. Néri, J.-L. Béchenec, F. Cappello, D. Etiemble (F)	19
Fault Tolerant Multiple-bus Interconnection Networks J. Sosnowski (P)	25
Experimental Evaluation of Mad Postman Bidimensional Routing Networks C. Izu (E), R. Beivide (E), C. Jesshope (UK) and A. Arruabarrena (E)	33
A2: Processor Architecture I. Chairman: K. Waldschmidt (D)	43
Evaluating Transport Triggered Architectures for Scalar Applications H. Corporaal (NL)	45
MOVE32INT, a Sea of Gates Realization of a High Performance Transport Triggered Architecture H. Corporaal and P. van der Arend (NL)	53
Design of a VLIW Architecture Constructed from Standard RISC Chips: A Case Study of Hardware/Software Codesign P. Pfahler, C. Nagel, F.-J. Rammig, U. Kastens (D)	61

A3:	System Specification I.	69
	Chairman: W. Grass (D)	
	X-Nets: a Visual Formalism for System Specification and Analysis	71
	S. Antoniazzi, A. Balboni and W. Fornaciari (I)	
	Automated Support for the Formal Specification and Design of Real-Time Systems	79
	C. Ho-Stuart, H. Zedan and M. Fang (UK)	
	SEAL. A Simple Language for Prototyping Action-event Specifications	87
	K. Brink, R. Huijsman and J. van Katwijk (NL)	
B1:	Distributed Memory Management	97
	Chairman: P. Milligan (UK)	
	The Memory Hierarchy of the CHESS Computer	99
	D. Lioupis, N. Kanellopoulos and M. Stefanidakis (GR)	
	Some Practical Considerations for the Implementer of the SCI Network	109
	V. Habchi, U. Finger and C. O'Donnell (F)	
	Conflict-free Access to Streams in Multiprocessor Systems	119
	M. Peiron, M. Valero, E. Ayguadé and T. Lang (E)	
B2:	Processor Architecture II.	131
	Chairman: J. Cortadella (E)	
	Evaluating the Cost of Conditional Branches on the Performance of Superscalar Machines	133
	E.S.T. Fernandes, F.M.B. Barbosa and D.M. Simpson (BR)	
	Branch Optimization of the TOOBSIE2 RISC-Processor and Classification	141
	M. Schäfers (D)	
	Arco: A Cost-Effective and Flexible Hardware Maze Router	149
	J.S. Aude, E.P.L. Filho, M.F. Martins and S.B. Pinto (BR)	
B3:	Network Management	161
	Chairman: J. Labarta (E)	
	Protection and Routing Algorithms for Network Management	163
	C.V. Papadopoulos and A.H. el Zahni (GR)	
	DIP: Distributed Diagnosis Protocol	171
	L.M. Silva and J.G. Silva (P)	

	PLANNER 4: A Simulator to Design Packet Switching Data Network S. Allegra, M. Annunziata, F. Cesaroni and C. Lunadei (I)	179
C1:	Parallel Software Engineering Chairman: P. Kacsuk (H)	187
	Tools For Object-Oriented SPMD Programming B. Moisan, Y. Duthen and R. Caubet (F)	189
	Distributed Active Objects: A Methodological Proposal and Tool for Distributed Programming with Transputer Systems M. Capel, J.M. Troya and A. Palma (E)	197
	A Reusable Load Balancer for Parallel Search Problems W. Joosen, S. Bijmens and P. Verbaeten (B)	205
	Real-time Behaviour Monitoring for Multi-processor Systems J.P. Calvez and O. Pasquier (F)	213
C2:	VLSI Testing and Testability Chairman: I. Erényi (H)	221
	Fault Behaviour and Testability of Asynchronous CMOS Circuits H.T. Vierhaus, W. Meyer, U. Gläser and R. Camposano (D)	223
	FSM Fault Models Impact on Test Performances C. Bolchini and F. Fummi (I)	229
	Redundancy Analysis and Removal for VLSI ASIC's K. Ivinskis (D)	237
	Memory Testing in a Massively Parallel Machine Ch. Aktouf, Ch. Robach and G. Mazaré (F)	245
C3:	Advances in Object-Oriented Design Chairman: K. Klöckner (D)	253
	Active Object Oriented Databases in Control Applications P. Loborg, T. Risch, M. Sköld and A. Törne (S)	255
	Heterogeneous Schema Integration Method for Multidatabase System Y. Lee and S. Moon (ROK)	265
	Toward Automated Support for Object-Oriented Modelling J. Ayre, D. Glass, J.G. Hughes and I.R. McChesney (UK)	273

	Incremental Resource Estimation with Real-Time Feedback from Measurement M. Oivo (SF)	281
C4:	Local Area Networks Chairman: H. Rzehak (D)	291
	Concurrent Ring: Design and Evaluation of a New Token Ring LAN J.D. Sandoval, F.A. Herrera, A. Suárez and C. Sandoval (E)	293
	A Distributed Bandwidth Allocation Algorithm for Gbit/s LANs T. Antonakopoulos, S. Koutroubinas, J. Koutsonikos and V. Makios (GR)	301
	An ISDN Primary Rate Interface for Ethernet Access F. Videira and A. Casaca (P)	309
D1:	Parallel Algorithms Chairman: E. Zapata (E)	317
	Parallel WZ Factorization on Mesh Multiprocessors R. Asenjo, M. Ujaldón and E.L. Zapata (E)	319
	Image Component Labelling on Reconfigurable Processor Array M. Maresca, P. Baglietto and A. Giordano (I)	327
	Computing the Generalized Eigenvalues of Symmetric Positive Definite Pencils on Networks of Transputers D. Giménez, V. Hernández and A.M. Vidal (E)	335
D2:	ASIC Design Chairman: N. Scarabottolo (I)	343
	Efficient Synthesis of Distributed Vector Multipliers B. Fehér (H)	345
	Implementation of Given's Rotation Processors for DSP Real-Time Applications D.E. Metafas, E.P. Mariatos, S.S. Nikolaidis and C.E. Goutis (GR)	351
	On Computing the 2-D Discrete Cosine Transform Using Rotations H.C. Karathanasis (GR)	359
D3:	Data Base Systems Chairman: A.M. Tyrrell (UK)	367
	Visual Query Language for Object-Oriented Databases: OOD J.C. Kwak and S. Moon (ROK)	369

	Global Query Management in Heterogeneous Distributed Database Systems S. Kang and S. Moon (ROK)	377
	A Distributed Deadlock Detection Algorithm for the AND Model J.R.G. de Mendívil, C.F. Alastruey and J.R. Garitagoitia (E)	385
D4:	Logic Design Chairman: A. Jávör (H)	393
	A Study of the Resetability of Synchronous Sequential Circuits A. Lioy and M. Poncino (I)	395
	Relational Algebra as Formalism for Hardware Design A.J.W.M. ten Berg, C. Huijs and Th. Krol (NL)	403
	An Object-Oriented Model of Design Evolution C. Charlton, P. Leng and M. Rivers (UK)	411
E1:	Parallelizing Compiler Techniques Chairman: R. McConnell (UK)	419
	Exact and Efficient Advanced Loop Interchange C-S. Yang, C-C. Chang, J-M. Yang, T-C. Huang and K.C. Huang (TW)	421
	Resource-Constrained Pipelining Based on Loop Transformations F. Sánchez and J. Cortadella (E)	429
	Optimal Data Dependence Chaining in Parallel Loops Z. Szczerbiński (P)	437
E2:	Microcontroller and Microprocessor Design A. Nunez (E)	445
	An Asynchronous Approach to the RISC Design of a Micro-Controller A. Costa, A. De Gloria, P. Faraboschi, G. Nateri and M. Olivieri (I)	447
	Application of the Novel Associative Programmable Array-structure Multi-Match-PLA in Synthesis of Decomposed Finite State Machines J. Kottsieper and K. Waldschmidt (D)	455
	OCEAN: A Hardware and Software Tool for Design of Future Smart Cards O. Caron, V. Cordonnier and G. Grimonprez (F)	467
E3:	Real-time Systems Chairman: J. Karjalainen (SF)	475

	Implementing and Checking Timing Constraints in Real-Time Programs K-J. Lin and K.B. Kenny (USA)	477
	On Guaranteeing Hard Real-Time Tasks C.E. Moron and H. Zedan (UK)	485
	Priority-Driven Concurrency Control Based on Data Conflict State in Distributed Real-Time Databases J. Kim and H. Shin (ROK)	491
E4:	VLSI Simulation Chairman: F. Distanto (I)	501
	SHC-SLX: A Levelized Compiled, Event Driven Interpreted VLSI Simulator L. Carro, C.A. Marcon and A.A. Suzim (BR)	503
	Timing Analysis for DCFL/SDCFL VLSI Circuits L. Gómez, A. Hernández and A. Nunez (E)	511
	Sequential and Parallel Strategies for the Demand-Driven Simulation of Logic Circuits P.E. Dunne, C.J.J. Gittings and P.H. Leng (UK)	519
F1:	Application-Driven Architectures Chairman: S. Winter (UK)	527
	A Processor Array Module for Distributed, Massively Parallel, Embedded Computing L. Bengtsson, K. Nilsson and B. Svensson (S)	529
	Design of an Extended Transputer Processor Farm System C.W. Yung and E.K.N. Yung (HK)	539
	A Parallel Architecture for the Color Doppler Flow Technique in Ultrasound Imaging A. Costa, A. De Gloria, P. Faraboschi and M. Olivieri (I)	545
	An Empirical Study of Automatic Restructuring of Vision Programs for SIMD Machines L.A. Barragán and A. Roy (E)	553
F2:	The FORMAT Project (ESPRIT 6028) Chairman: J. Goicolea (E)	561
	A Design Methodology for the Correct Specification of VLSI Systems C. Bolchini, M. Bombana, P. Cavalloro, C. Costi, F. Fummi and G. Zaza (I)	563

	VHDL/S - Integrating Statecharts, Timing Diagrams, and VHDL J. Helbig, R. Schlör, W. Damm, G. Döhmen and P. Kelb (D)	571
	Introducing Structure into Behavioural Descriptions Obtained from Timing Diagram Specifications W.D. Tiedemann, S. Lenk, C. Grobe and W. Grass (D)	581
	VHDL Generation from a Timed Extension of the Formal Description Technique LOTOS within the FORMAT Project C.D. Kloos, T. de Miguel Moro, T.R. Valladares, G.R. Filho and A.M. López (E)	589
	Analysis Tools Applied to VHDL S. Olcoz and J.M. Colom (E)	597
F3:	From Requirements to Implementation Chairman: M.J. Taylor (UK)	605
	Real-Time Specification Using Petri Nets K. Sacha (PL)	607
	IDFG: An Interactive Applications Specification Model with Phenomenological Properties A. Kameas, S. Papadimitriou, P. Pintelas and G. Pavlides (GR)	615
	Three Approaches to Interprocedural Dynamic Slicing M. Kamkar, P. Fritzson and N. Shahmehri (S)	625
	A Software System for the Generation of Test Data for Ada Programs M.J. Gallagher and V.L. Narasimhan (AUS)	637
F4:	Issues in Distributed Systems Chairman: K.P. Judmann (A)	645
	Concurrency Control Algorithms for Real-Time Systems H. Nakazato and K-J. Lin (USA)	647
	Analysis and Simulation of Six Bus Arbitration Protocols V.L. Narasimhan and S. Price-White (AUS)	655
	Design of TMS320C40 Signal Processors and Programmable Logic Based Prototyping Environment of Real-time Machine Vision Architectures K. Rautiola, K. Pehkonen, L. Stähle and P. Jokitalo (SF)	663
	Modelling and Design of the Multimedia Subsystem of a Distributed Authoring Environment A. Kameas, S. Papadimitriou and P. Pintelas (GR)	669

G1:	Neural Networks G. Kock (D)	679
	Mapping and Parallel Simulation of Synchronous Neural Networks on Multiprocessors T. Reski (D)	681
	Neural Accelerator for Parallelization of Back-Propagation Algorithm E. Franzi (CH)	689
	Removal of Catastrophic Noise in Hetero-Associative Training Samples E. Tuv and A.N. Refenes (UK)	697
G2:	Algorithms and Tools for VLSI Design Chairman: L. Jóźwiak (NL)	705
	An Empirical Study of Genetic Operators in Genetic Algorithms Xin Yao (AUS)	707
	Automatic Test Pattern Generation can be Solved as a Constraint Satisfaction Problem K. Tilly, L. Surján and G. Román (H)	715
	A Behavioral Level Tool for Design and Verification of Parallel Architectures: PADS F. Distanto and P. Toccaceli (I)	723
G3:	Protocols and Performance Analysis Chairman: G. Ciccarella (I)	731
	Providing Circuit Service over a High Speed Deflection Network E. Pagani and G.P. Rossi (I)	733
	Dynamic Protocol Configuration for Multimedia Networks R. Stainov (D)	741
	Performance Analysis of Synchronous Multibuffered Packet-Switching Networks A. Tentov and A. Grnarov (MC)	749
G4:	RT-level Synthesis Chairman: F. Tirado (E)	757
	Automata Decomposition and Synthesis with PLAM S. Baranov and L. Bregman (IL)	759

	Synthesis for Testability by Synthesis Controlling C. Nagel (D)	767
	GLASS: a Graph-Theoretical Approach for Global Binding R.M. Badia and J. Cortadella (E)	775
H1:	Fault Tolerance Chairman: K.E. Grosspietsch (D)	783
	The Design of Time-critical Conversations A.M. Tyrrell and G.F. Carpenter (UK)	785
	A Fault-Tolerant Server on MACH S. Arévalo, J. Carretero, J.L. Castellanos and F. Barco (E)	793
H2:	Decision Making Under Uncertainty Chairman: A. Rea (UK)	801
	Pen-based Recognizing of Handprinted Characters B. Klauer and K. Waldschmidt (D)	803
	N Version Programming with Majority Voting Decision: Dependability Modelling and Evaluation K. Goševa-Popstojanova and A. Grnarov (MC)	811
H3:	FDT Tools Chairman: H. Schumny (D)	819
	PROCONSUL, a Tool for Computer-aided Protocol Engineering K. Tarnay, Gy. Csopaki, M. Törő, M.D. Ta, Zs.B. Harangozó and J. Miskolczi (H)	821
	Test Strategy for Indeterministic Protocols M. Törő (H)	827
H4:	System Specification II. Chairman: D. Sanders (UK)	833
	TL: A System Specification System L.P.M. Benders and M.P.J. Stevens (NL)	835
	Connectionist Model Description: a Case Study G. Kock and N.B. Šerbedžija (D)	843
	Session Chairpersons' addresses	851
	Author index	855
	Subject index	859

DIP : Distributed Diagnosis Protocol

Luís Moura Silva and João Gabriel Silva

Laboratório de Informática e Sistemas
Universidade de Coimbra
Urb. Boavista, Lt. 1-1
3000 Coimbra - Portugal
Fax : +351.39.701266
e-mail : lams@uc.pt

Abstract

This paper presents the DIP - a Diagnosis Protocol to check the availability of processors in a distributed system. Each node maintains a Network Status Table, and failure detection is achieved by alive messages. The overhead of such messages is minimal and the main virtue of the protocol is the lower diagnosis latency when a failure is detected. Since the unavailability of a node can be the result of a processor crash or a network partition, the DIP protocol is also able to cope with partitions. The protocol can diagnose fail-stop and omission failures. Moreover, it is robust against out-of-order messages, duplicates and false-alarms.

Keywords : *Distributed Diagnosis, Fault Tolerance, Membership Protocols, Network Monitoring.*

1. Introduction

One of the requirements of a fault-tolerant distributed system is the failure detection capability of remote sites. Recovery and reconfiguration are other desirable properties. Since the paper of Preparata, Metze and Chien [1] there were further theoretical developments [2] in the area of distributed diagnosis. As a result, several PMC-based diagnosis algorithms have been suggested [3-9]. Recently, Bianchini *et al.* [10,11] have conducted the first implementations of system diagnosis theory in a real distributed network environment. Nevertheless, all those PMC-based algorithms have serious drawbacks:

1) when the number of node failures exceeds the connectivity of the system digraph, those algorithms simply yield false diagnostic results. 2) The PMC-model is only valid for permanent processor failures and it assumes that a node cannot fail and restart in the period between two tests. 3) Most of the algorithms require a high number of diagnostic messages and, as a result, the execution time may exceed reasonable values.

All these shortcomings make the algorithms unsuited to local area networks. Moreover, if one considers that processors may exhibit byzantine faultiness [12], PMC-based algorithms may identify

none of the faulty processors [13]. Another class of diagnosis algorithms which identifies byzantine faults has been suggested [13-15]. These algorithms are able to diagnose intermittent and malicious faults. However, the diagnosis of byzantine faults becomes extremely difficult and all the proposed solutions present a great message overhead. As practitioners, we are not willing to pay such price and we simply assume a more restricted failure model: the crash failure model. In this model it is assumed that each node can only fail in "clean" well defined ways. During its functioning the node behaves correctly and if an internal failure occurs it halts automatically. Such nodes are called fail-silent or fail-stop. A lot of algorithms [18-27] have also assumed this failure model but no one seems to perfectly suit our goals and assumptions.

2. System Model and Assumptions

The system consists of a set of N stations interconnected by a LAN with broadcast capability. Each station has access to *stable storage* which can be used to store information that must survive to failures. In perfect conditions each processor can communicate with any other processor in the network. We assume that the relation "can-communicate" between nodes is symmetric and

transitive. Two kinds of failures are considered: processor failures and communication link failures. Processor failures are assumed to be clean. When a node fails it simply stops running and does not send any arbitrary or malicious messages. Thus, the universe of failures includes stop and omission failures and no assumption is made on the number of allowable processor failures.

Network failures can be classified in hard and soft failures [16]. An hard failure is characterized by the incapability of the network to deliver any bandwidth at all. Soft failures are characterized by performance degradations or deviations from the expected behaviour. In networks, such as the Ethernet [17], which do not provide any hardware level acknowledgment, the other nodes of a network cannot distinguish an isolated node from a crashed one, since in both cases the symptom is the same - the loss of communication with the node. When an active node is not available to any other node, it means that the network was partitioned (hard failure) and both nodes are in different partitions. However, crashed nodes can recover and partitions can be reconnected. For the sake of generality let us consider the existence of two main events : crashes and restarts. Each available node is marked as UP while an unavailable node is classified as DOWN. Each processor will have its own view of the network which must be consistent with that of other nodes belonging to the same partition.

We assume that the message delivery is not reliable since the network is subjected to soft failures. Messages can be lost, delayed or delivered out-of-order. However, a network soft failure cannot produce any malicious message.

We consider that there exists a maximum message delay $\partial_{\max}$, which reflects the expected maximum transmission time of a single message between two nodes, including the processing delay. The minimum message delay is represented by $\partial_{\min}$. Every node should have a physical clock in order to implement timers but we do not require clocks to be synchronized.

3. Design Guidelines

The DIP protocol should be able to perform three functions :

- (1) To detect node crashes.
- (2) To notify all the network nodes of the crash of a processor (Node Down event).

(3) and also to notify all the nodes of the restart of a crashed node (Node Up event).

Each node maintains a table with the status of the other nodes, called the *Network Status Table* (NST). Each entry of the NST has three fields : the host name, its status and the *Incarnation Number* (IN). The IN number is helpful to distinguish duplicates and out-of-order messages, as will be explained in §4.4.

Since failures are the exception rather than the rule, the first imperative is to reduce the overhead during normal operation. The protocol must assure a minimal number of messages during steady-state operation. Another important goal is to reduce the diagnosis latency to a minimal value, i.e., the protocol must provide timely failure detection. This is a paramount figure of merit to allow an effective recovery/reconfiguration of distributed applications. When a node failure occurs all the other nodes must be acquainted with that as quickly as possible. As failures and restarts are asynchronous events, the protocol should be robust against their occurrence during its execution. Other important issue is the handling of network partitions. The other desired goals are efficiency, simplicity and correctness of the protocol.

4. Design Details

4.1 The Detection of Failures

As a fail-silent processor only stops when an internal fault occurs the failure can be simply detected by background *I_AM_ALIVE* messages. The network nodes are organized on a virtual ring and all the sites are numbered from 0 up to $N-1$, along the path of the virtual ring. It should be noted that the ring does not reflect the physical organization of the network. The order of the nodes in the ring follows the order of the Status Table, which is unique to all the processors in the network.

Each processor P_i sends periodically an *I_AM_ALIVE* message to the next neighbour $P_{((i+1) \bmod N)}$ while it is checking for the messages sent by the previous processor $P_{((i-1) \bmod N)}$. Each processor has two independent processes to perform that job.

- 1) The *Heartbeat* process is periodically awakened by a timer interrupt in order to send a message to the next neighbour. This timer is represented by T_H and is set to α .

2) The detection of failures is done by the Watchdog process, which is listening to the messages sent by the previous neighbour. Every time it receives an `I_AM_ALIVE` message it triggers a timer T_W with $((\Omega+1) \cdot \alpha + \partial_{\max} - \partial_{\min})$, where Ω is the number of omission failures or lost messages that can be tolerated. If the timer expires, this means that the previous node has crashed and the Node Down Protocol is started for that node.

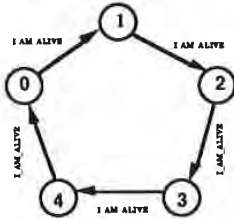


Figure 1 : Virtual Ring

4.2 Node Down Protocol

In order to reduce the diagnosis latency, the node should notify all the others immediately by a broadcast message to the network, announcing the `Node_Down` event of its neighbour.

Take as an example the network of figure 2. One can notice that node #2 has crashed and its tester (node #3) will detect that when its timer T_W expires. The status of node #2 is marked as DOWN in its Network Status Table, and it will send a broadcast message to the network, which includes the number of the node and the `NST(2).IN : NODE_DOWN(2, 1)`.

In a meantime, a timer T_{ack} will be set to $2 \cdot \partial_{\max}$. The node will inspect its status table to find the available nodes and waits for the acknowledge of every UP node. On the other hand, when a node receives a `NODE_DOWN` broadcast it replies with a `ND_Ack` message to node #3. It should be noted that each acknowledge is also a broadcast message. This may be useful if a particular node does not receive the broadcast message of #3 it may be informed of that by one of the acknowledge messages. When node #3 receives all acknowledges this phase is finished and the protocol returns to the steady state. However, if the timer T_{ack} expires and some reply is missing, the message `NODE_DOWN(2, 1)` is retransmitted again (until a maximum number of retries is reached).

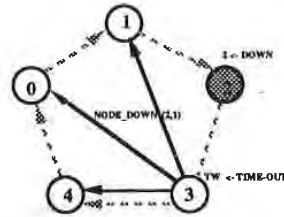


Figure 2 : Detection of a Failure

It may happen that the unavailability of node #2 is not the crash of a single processor, but instead the network may be partitioned. In this case, some nodes may not answer to the message and should also be classified as DOWN. In fact, if after the retransmission, node #3 does not receive any reply then it will start again this protocol announcing the unavailability of those nodes which remained in the other partition of the network.

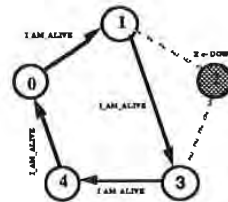


Figure 3 : Reconfiguration

After the notification phase of a node crash, the testing structure has to be adapted to a new configuration. Each node that had been notified that its tester neighbour has crashed must identify the next available node, by searching the Status Table (see example of node #1 in figure 3). The detector node (#3) will also have a new tested node (#1).

4.3 Node Up Protocol

When a node rejoins the network it must tell the others that it is back in operation and must also determine which nodes are UP and reachable. Therefore, when a node recovers it reads the NST from stable storage to main memory. The status of all the other nodes are set to DOWN. Then it increments its IN value, which is also stored in stable storage, and sets to zero all the counters of the other nodes. Finally, it broadcasts a `NODE_UP` message to notify all the active nodes in the

network. A timer T_{up} is activated. This message must be acknowledged by the other available nodes. The replies will be used by the restarted node to achieve its new network view. Furthermore, to guarantee a distributed consistency, node #2 must verify whether all the nodes it can reach share the same NST.

Let us consider the previous scenario : node #2 is DOWN and its IN is equal to 1. Now, suppose that it recovers and rejoins the network. First of all, its IN counter will be incremented, and it will send a `NODE_UP` broadcast message to the network, which includes the new IN (see figure 4).

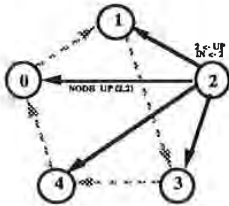


Figure 4 : Node #2 Rejoins the Network

When each of the other nodes receive the `NODE_UP(2,2)` message they check the 2nd entry of its NST, and the status of node #2 is marked as UP. Then, they send a `NU_Ack` message to node #2, which contains their own INs. Each node also appends a list of its DOWN nodes into that message, including the corresponding INs. When the timer T_{up} expires, node #2 will have a complete information about the network status. Likewise, to assure a distributed agreement, node #2 must check the information it had received from the `NU_Ack` messages. The DOWN lists which were piggybacked into the acknowledges must be all consistent. Otherwise, node #2 will have to broadcast the result of the comparison to force a distributed consensus.

4.4 Protocol Integration

As the two sub-protocols may execute simultaneously they must interact with each other. Let us consider the following situation : node #2 has crashed ; node #3 detects its failure and sends a `NODE_DOWN` broadcast message to the network. However node #2 is DOWN for only a short time, after that it comes up again and rejoins the network by sending a `NODE_UP` message. Yet, an

inconsistent scenario may occur: Node #0 receives the sequence (`NODE_DOWN`, `NODE_UP`) while node #1 receives the (`NODE_UP`, `NODE_DOWN`) sequence. Fortunately, the *Incarnation Number* can solve this inconsistency. Every message is tagged with the IN and each receptive node will compare the incoming IN with the value stored in its NST. When a host receives a `NODE_UP` message from node #K and the incoming IN exceeds $NST(K).IN$, the table is automatically updated. If the incoming IN is similar it is recognized as a duplicate message and no action is performed. In either cases a reply (`NU_Ack`) is sent. When a node receives a `NODE_DOWN` message from node #X notifying the crash of its neighbour (node #Y) it checks in its table the status and the current IN of node #Y. If #Y is UP and the incoming IN exceeds the $NST(Y).IN$, then #Y is marked as DOWN and the local IN is updated. In normal situations, the incoming IN is similar to the local value. In this case, only the status of node #Y is updated. On the other hand, if #Y is already DOWN the $NST(Y).IN$ will only be updated if the incoming IN exceeds the local IN. In either case the receiving node will reply to each message with its own IN.

Taking the previous example, let us assume that node #2 was running with $IN=1$. By the time it had crashed, node #3 would have sent a `NODE_DOWN` message announcing this event (with $IN=1$). When the node recovers, it would increment IN to 2.

Node #0 would receive the messages in the correct order. However, when node #1 received the message `NODE_UP(2,2)` it would realize that node #2 had crashed and came up again without being aware of that. It would set $NST(2).IN$ to 2 and send a `NU_Ack` message to node #2, which would include its own IN ($NST(1).IN$).

After having received the `NODE_DOWN(2,1)` message it would conclude that the message was out-of-date since the incoming IN was less than the stored one. In spite of that, a reply message (`ND_Ack`) would be sent to node #3.

4.5 False Alarms

A *false alarm* occurs when an active node receives a `NODE_DOWN` broadcast message notifying its own failure. It may happen if the number of lost messages exceeds the tolerated level. A network soft failure can be the cause of that. In this case, the node will react to that "injury" by broadcasting a

I'M_NOT_DOWN message to the network with its current IN. When the other nodes receive such message they undo the action taken by the reception of the *false alarm* message. This message is only taken into account if the incoming IN is equal to the stored value.

4.6 Coping with Network Partitions

Network partitioning is an event which causes the network to divide into two or more sub-networks. The worst case is when a single node is isolated from the rest of the network due to a crash of its communication processor. Partitioning should be detected during the execution of the Node Down Protocol.

To simplify the explanation let us consider the network represented in figure 5. Due to a communication link breakdown or to some other network component (e.g. gateway, bridge, router...) failure, the network was split into two sub-networks. The detection of this partitioning begins with the detection of two single node failures. Node #2 and node #0 will detect the crashes of node #1 and node #4, respectively. Each of the detectors will send a NODE_DOWN broadcast and then wait for the acknowledge messages. As Node #2 will note the lack of response from node #0 it will repeat the Down protocol for node #0. On the other hand, this node only receives the ND_Ack from node #1. Thus, it will broadcast to its sub-network the failure of node #2 and #3. Only node #1 will hear these messages.

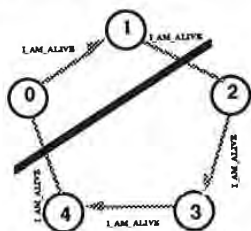


Figure 5 : Network Partitioning

The final result of this phase is represented in figure 6. Two partitions have been identified (P1 and P2) and the main graph of test has been divided into two independent graphs. To the nodes of partition P1, the nodes belonging to P2 are marked as DOWN, and vice-versa. The protocol guarantees that each time a network partitioning is detected this is assumed at

both sides of the "wall" between partitions.

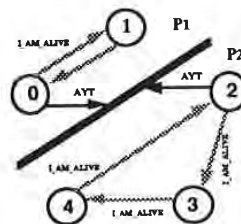


Figure 6 : Partitioning Reconfiguration

Furthermore, it is necessary to assure the detection of the possible merge of these partitions. This job is assigned to both *detector* nodes (#0 and #2). Each one will resume a *dummy* process that periodically sends a AYT² broadcast message to the network, which contains the identification of the sender and its IN. A rule is established to the nodes of the same partition : they do not answer to those messages. For instance, in partition P1 node #0 sends periodically an AYT broadcast message. Since node #0 is UP for node #1, it will not reply.

Those messages may pass through the partition "wall", and can be heard by the other nodes only when the two sub-networks get merged. Each node that receives an AYT message from a sender judged as DOWN, will acknowledge it with a YIA³ message. This reply contains the node identification, its current IN and is also broadcasted. A YIA works like a NODE_UP message with just one exception : if the incoming IN is similar to or exceeds the local IN, the message is accepted and the table is updated for the sender of the message. If the incoming IN is smaller, it is classified as obsolete. In both cases, a reply is sent to the network.

In the previous example, nodes #2 and #0 kept watching the other partition. They periodically send a AYT messages to the network. While the partition is maintained, those messages are not acknowledged. When the clusters get reconnected, node #1 and #0 hear the AYT(2) while nodes #2, #3 and #4 hear the AYT(0) message. They answer to those messages with a broadcast YIA and a special Node_Up protocol is started in each node. After that phase, the network recovers to the original configuration.

² AYT means "Are You There?"

³ YIA means "Yes, I Am !"

As it is not possible to distinguish a node failure from a network partition, every time a node detects a node crash it resumes that *dummy* process, which is suspended when all the failed nodes it has detected get up again. As it is represented in figure 7 it may happen that network partitioning does not affect a contiguous set of nodes (as in fig.6), but instead an alternate set along the path of the virtual ring. This is the worst case in terms of the number of messages, as will be seen in the next section.

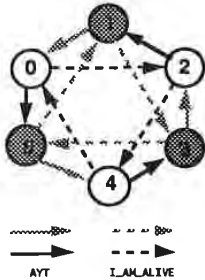


Figure 7 : Special Partitioning

5. Analysis of the Protocol

The original protocol uses 8 different messages: *I_AM_ALIVE*, *NODE_UP*, *NU_Ack*, *NODE_DOWN*, *ND_Ack*, *I'M_NOT_DOWN*, *AYT*, *YIA*.

In steady-state operation, the protocol requires the minimum number of diagnostic messages. In a network of N active processors, N messages are sent to the network between each testing interval. When a node failure occurs, it is detected by its neighbour in a finite interval called *failure detection latency*. This latency depends on how much time has elapsed between the time the last *I_AM_ALIVE* message was sent and the failure. In the worst case, the failure occurs immediately after the last message is sent out, while in the best case the node fails just before sending the message. So, the detection latency falls between (∂_{max}) and the period of the Watchdog Timer: $((\Omega+1) \cdot \alpha + \partial_{max} - \partial_{min})$.

Let us now refer to the time that it takes to notify the failure to all the other nodes the *diagnosis latency*. In the DIP approach, the path length of the diagnosis information is equal to 1 and the maximum diagnosis latency is just ∂_{max} . The diagnosability of the protocol is equal to $(N-1)$, and the number of messages necessary to perform the *Node_Down* protocol is bound to $(N-1)$ (1 broadcast

and $(N-2)$ replies)⁴. To notify a *Node_Up* event it only requires N messages. However, if we have a network with (S) subnets connected through routers, the protocol has to send (S) *NODE_UP* messages, one for each subnet, and the diagnosis latency is also increased by the processing time of each intermediate router. The message count will be bound to $(S+N-2)$ for the *Node_Down* Protocol, while the *Node_Up* protocol will use $(S+N-1)$ messages.

Now, let us consider the partitioning of the network into P partitions. If the set of nodes of each partition is contiguous as in figure 6 the number of messages will be $(P+N) : (P)$ *AYT* messages and (N) *I_AM_ALIVE* messages. In the worst case, as shown in figure 7, the message count will be bound to $(P \cdot N)$. When only one node has become isolated, the number of messages will be $(N+1)$.

6. Relation to Previous Work

In this section, we will briefly discuss others algorithms found in the literature, and a comparison with the DIP will be made. The implementation of the *NEW-SELF* algorithm, conducted by Bianchini *et al.* [10] has shown that it requires a high number of diagnostic messages. It was concluded that such algorithm is not feasible to real systems. In the same paper, a new algorithm is proposed: the *EVENT-SELF*. This one reduces the message count during normal operation but if a failure occurs it produces a packet count higher than the one of *NEW-SELF*. Moreover, an experiment with 20 processors has shown that the complete diagnosis of one processor failure would take about 154 seconds. Both algorithms have a static testing graph, which means that the number of faulty units is bounded to a well established limit (t) . If the number of processor failures exceeds that number (t) , the network becomes undiagnosable.

In order to solve these serious problems, Bianchini proposed a new algorithm: the *Adaptive DSD* [11]. This algorithm can diagnose any number of failures since it has an adaptive testing graph. This represents some improvements, but it also has some shortcomings. Since it is a *PMC*-based algorithm it does not support that a node fails and recovers between two tests executed by another node. The algorithm is not able to diagnose link failures, and the path length of the diagnosis information is bound

⁴ Assuming no retransmissions.

to $\log_2 N$. An experiment in a network of 60 processors with a version that uses an asymmetric forwarding scheme has shown that the overall diagnosis latency was about 90 seconds for one processor failure. The version that assumes the original model of forwarding of the PMC-model has an overall diagnosis latency of 450 seconds.

If one wants an efficient algorithm the latency should be drastically reduced and the number of messages minimized. The algorithms that assume the crash failure model are the most suitable and can be easily applied in real systems since they require a lower message count and are much more simple.

A centralized approach was chosen in the Publishing system [18]. In this case, a dedicated node, called the recorder, has a watchdog process for each processor of the network. If no messages are received in a time limit, the processor is considered to have crashed. This approach is simple but has a weak point in the recording node. Another centralized approach was chosen in [19], which uses an election algorithm to choose the master node.

At the other extreme, we found the detection mechanism of the TANDEM NonStop system [20], where each node tests every other node in the network. The DIP approach falls between the centralized and the fully distributed approach.

A practical algorithm can be found in [21]. It has some similarities with the DIP. However, that proposal has some shortcomings. First, it leaves the failure detection job to the application. Second, when a node fails, the number of diagnosis messages is $2N^2 - 6N + 4$ (where N is the number of nodes). This means that the network will be flooded with messages, and the diagnosis latency will suffer from that.

The algorithm presented by Walter [22] is similar to the DIP - it also organizes the network as a virtual ring of processors. But, one may notice that it is not robust against lost messages, duplicates, out-of-order messages and false alarms, since it assumes a reliable network. The critical message path of the diagnosis is 3, while the critical path of the DIP is just 1, which means that the diagnosis latency is at least 3 times higher than the latency of DIP. During steady-state operation the RSC protocol always requires the same number of messages: $P \cdot N$.

Another algorithm implemented in the SDD/1's RelNet [23] provides a systematic detection of failures and node recoveries. Nevertheless, that

algorithm is not robust against network partitions. Another set of algorithms has been proposed for synchronous distributed systems [24-27].

7. Implementation Features

The first release of the DIP was implemented on a set of SUN workstations which communicate through an Ethernet local area network. It was written in C using sockets UDP/IP. The protocol itself was running as a background process. The initialization module stores in stable storage the list of the processors in the network. This list is sorted by the Internet host address. As this number is unique to each processor it assures that all the hosts generate identical lists. All the crash and recovery events are stored in the *Crash_Log* file, which can be inspected for failure analysis. A query interface is also available to users asking for the network status. The parameters chosen were the following: $\alpha = 1$ second and $\Omega = 2$.

The DIP provides a failure detection capability to be incorporated in a recovery mechanism, based on a distributed checkpointing algorithm presented elsewhere [28].

8. Conclusions and Future Work

We have presented a protocol for checking the availability of sites in local area networks. During steady-state operation it uses the minimum number of messages, while in case of failure it assures the lowest diagnosis latency. Its testing graph is adaptive which provides a complete diagnosability. It is also robust against false alarms, out-of-order messages and network partitions. The only shortcoming is its reduced failure model. If anyone wants to extend the model should try one of the techniques presented by Neiger and Toueg [29] to automatically translate this algorithm, that only tolerates crash failures, into another one that tolerates more "severe" failures (i.e. arbitrary or byzantine). As future work we are planning to implement the protocol on a network of transputers.

Acknowledgments

L.M.Silva was supported by JNICT under the Programa Ciência (BM-38-90-IA). This work was partially supported by INIC and by JNICT under contract PMCT/C/TIT/451/90.

References

1. F.P.Preparata, G.Metze and R.T.Chien. "On the connection assignment problem of diagnosable systems", IEEE Transactions on Electronic Computing, 16, Dec.1967, pp.848-854
2. S.E.Kreutzer and S.L.Hakimi. "System - Level Fault Diagnosis : A Survey", Euromicro Journal, May 1987, pp.323-330
3. J.G.Kuhl and S.M.Reddy. "Distributed Fault-Tolerance for Large Multi-Processor Systems", Proc. 7th Annual Symposium on Computer Architectures, May 1980, pp. 23-30
4. J.G.Kuhl and S.M.Reddy. "Fault-Diagnosis in Fully Distributed Systems", Proc. FTCS-11, 1981 pp. 100-105
5. S.H.Hosseini, J.G.Kuhl and S.M.Reddy. "On Self-Diagnosis of the Distributed Systems", Proc. FTCS-15, 1985, pp. 30-35
6. S.H.Hosseini, J.G.Kuhl and S.M.Reddy. "A Diagnosis Algorithm for Distributed Computing Systems with Dynamic Failure and Repair", IEEE Trans.on Computers, March 1984, pp.223-233
7. S.E.Kreutzer and S.L.Hakimi. "Distributed Diagnosis and the System User", IEEE Trans. on Computers, January 1988, pp. 71-78
8. N.H. Vaidja and D.K. Pradhan. "System Level Diagnosis : Combining Detection and Location", Proc. FTCS-21, 1991, pp.488-495
9. A.Baghi and S.L.Hakimi. "An Optimal Algorithm for Distributed System Level Diagnosis", Proc. FTCS-21, 1991, pp. 214-221
10. R.Bianchini, K.Goodwin and D.S.Nydick. "Practical Application and Implementation of Distributed System-Level Diagnosis Theory" Proc. FTCS-20, 1990, pp. 332-339
11. R.Bianchini and R.Buskens. "An Adaptive Distributed System-Level Diagnosis Algorithm and Its Implementation", Proc. FTCS-21, 1991, pp. 222-229
12. L.Lamport, R.Shostak and M.Pease. "The Byzantine Generals Problem", ACM Trans. on Programming Languages and Systems, V. 4 1982, pp.382-401
13. K.V.S.Ramarao and J.C.Adams. "On the Diagnosis of Byzantine Faults", Proc. of the 7th Symposium on Reliable Distributed Systems, 1988, pp.144-153
14. K.C.Shin and P.Ramanathan. "Diagnosis of Processors with Byzantine Faults in a Distributed Computing System", Proc. FTCS-17, 1987, pp.55-60
15. J.C.Adams and K.V.S. Ramarao. "Distributed Diagnosis of Byzantine Processors and Links", Proc. 9th Int. Conf. on Distributed Computing Systems, 1989, pp.562-569
16. R.A. Maxion and F.E. Feather. "A Case Study of Ethernet Anomalies in a Distributed Computing Environment", IEEE Transactions on Reliability, Vol 39, October 1990, pp.433-443
17. J.F. Schock et al. "Evolution of the Ethernet Local Computer Network", IEEE Computer, August 1982, Vol. 5 (8), pp. 10-27
18. M.L.Powell and D.L.Presoto. "Publishing : A Reliable Broadcast Communication Mechanism", Proc. 9th Symposium on Operating Systems Principles, 1983, pp.100-109
19. T.Becker. "Keeping Processes Under Surveillance", Proc. 10th Symposium on Reliable Distributed Systems, 1991, pp.198-205
20. C.I. Dimmer. "The Tandem Non-Stop System", in *Resilient Computing Systems*, Ed. by T. Anderson, 1985, pp. 178-196
21. S.Bruso. "A Failure Detection and Notification Protocol for Distributed Computing Systems", Proc. 5th Int. Conf. on Distributed Computing Systems, 1985, pp.116-123
22. B.Walter. "A Robust and Efficient Protocol for Checking the Availability of Remote Sites", Computer Networks, Vol. 6, 1982, pp. 173-188
23. M.Hammer and D.Shipman. "Reliability Mechanisms for SDD-1 : A System for Distributed Databases", ACM Transactions on Database Systems, Vol.5 (4), Dec. 1980, pp. 431-466
24. F.Cristian. "Agreeing on Who Is Present and Who Is Absent In a Synchronous Distributed System" Proc. FTCS-18, 1988, pp. 206-211
25. H.Kopetz, G.Grunsteidl and J.Reisinger. "Fault-Tolerant Membership Service in a Synchronous Distributed Real-Time System", in *Dependable Computing for Critical Applications*, Springer Verlag, 1991
26. B.Walter. "Network Partitioning and Symmetric Surveillance Protocols", Proc. 5th Int. Conf. on Distributed Computing Systems, 85, pp.124 -129
27. A.V.Ma and G.G.Belford. "A Failure and Recovery Detection Protocol for Optimistic Partitioned Operation in Distributed Database Systems", Proc. 6th Int. Conf. on Distributed Computing Systems, 1986, pp.540-547
28. L.M.Silva, J.G. Silva. "Global Checkpointing for Distributed Programs", Proc. 11th Symp. on Reliable Distributed Systems, pp. 155-162, 1992, Houston (USA)
29. G.Neiger and S.Toueg. "Automatically Increasing the Fault-Tolerance of Distributed Systems", Proc. 7th Symp. on Principles of Distributed Computing, 1988, pp.248-262

APPENDIX 1005B

LIBRARY | CATALOG

← < 1 of 1 >

PERIODICAL OR NEWSPAPER

Microprocessing and microprogramming

Full Record **MARC Tags**

Main title

Microprocessing and microprogramming.

Published/Created

Amsterdam : North-Holland Pub. Co., 1981-[1996]

Request this Item

 LC Find It

More Information >

LCCN Permalink <https://lcn.loc.gov/86641015>

Publication history Vol. 7, no. 1 (Jan. 1981)-v. 41, no. 10 (June 1996).

Description 35 volumes : illustrations ; 27 cm

Current frequency Twelve no. a year, 1994-

Former frequency Ten no. a year, 1981-1986
Fifteen no. a year, 1987-1993

ISSN 0165-6074

Linking ISSN 0165-6074

LC classification	QA76.5 .E89a
Continues	Euromicro journal 0167-3858 (DLC) 78644982 (OCoLC)4109280
Continued by	Journal of systems architecture 1383-7621 (DLC) 96658139 (OCoLC)35360253
Serial key title	Microprocessing and microprogramming
Abbreviated title	Microprocess. microprogram.
Related names	Euromicro.
LC Subjects	Microprocessors--Periodicals. Microprogramming--Periodicals.
Other Subjects	Microprocessors. Microprogramming. MICROPROGRAMMING. DATA PROCESSING.
Form/Genre	Electronic journals. Periodicals.
Browse by shelf order	QA76.5
Notes	Published for: Euromicro.
Additional formats	Online version: Microprocessing and microprogramming (Online) (DLC) 2005233216 (OCoLC)41227882 Online version: Microprocessing and microprogramming (OCoLC)564648579 Online version: Microprocessing and microprogramming (OCoLC)631723842
LCCN	86641015
Invalid LCCN	sn 82020408
CODEN	MMICDT
Dewey class no.	004.16
National bib agency no.	007522784 B8163166

Other system no.	(OCoLC)ocm07324063
Additional Links	http://www.sciencedirect.com/science/journal/01656074
Type of material	Periodical or Newspaper
Content type	text
Media type	unmediated
Carrier type	volume

Item Availability



CALL NUMBER	QA76.5 .E89a
Request in	Jefferson or Adams Building Reading Rooms
Status	Not Charged

CALL NUMBER	QA76.5 .E89a Copy 1
Request in	Jefferson or Adams Building Reading Rooms
Status	Not Charged

Older receipts	13-14 * 1984 : Feb.-Dec. 15-16 * 1985 17-18 * 1986 19 * 1987 : Jan. 31-32 * 1991 : Apr.-Aug. 33-34 * 1991 -92: Sept.-Feb. 35 * 1992 : Sept. 36-37 * 1992 -93: Nov.-Jan. 38-39 * 1993 : Sept.-Dec. 40 * 1994 41 APR-NOV * 1995 7:1 * 1981 : Jan.
----------------	--

LIBRARY

CATALOG



PERIODICAL OR NEWSPAPER

Microprocessing and microprogramming

[Full Record](#)[MARC Tags](#)

000 03706cas a2200757 a 4500
001 11312143
005 20210819065825.0
008 810410d19811996ne mx p 0 a0eng c
010 __ |a 86641015 |z sn 82020408
016 7_ |a 007522784 |2 Uk
016 7_ |a B8163166 |2 bccb
022 __ |a 0165-6074 |l 0165-6074 |2 z
030 __ |a MMICDT
035 __ |a (OCoLC)ocm07324063
040 __ |a NDD |b eng |c NDD |d IXA |d NST |d AIP |d NST |d AIP |d DLC |d NST |d NSD |d NST |d NSD
|d UMC |d WAU |d DLC |d OCLCQ |d CUS |d GUA |d NLGGC |d NYDWH |d OCLCQ |d UKMGB |
|d OCLCF |d IUL |d AU@ |d OCLCQ
042 __ |a pcc
050 00 |a QA76.5 |b .E89a
082 0_ |a 004.16 |2 19
210 0_ |a Microprocess. microprogram.
222 _0 |a Microprocessing and microprogramming
245 00 |a Microprocessing and microprogramming.
260 __ |a Amsterdam : |b North-Holland Pub. Co., |c 1981-[1996]
300 __ |a 35 volumes : |b illustrations ; |c 27 cm
310 __ |a Twelve no. a year, |b 1994-
321 __ |a Ten no. a year, |b 1981-1986
321 __ |a Fifteen no. a year, |b 1987-1993

336 ___ |a text |b txt |2 rdacontent
 337 ___ |a unmediated |b n |2 rdamedia
 338 ___ |a volume |b nc |2 rdacarrier
 362 0_ |a Vol. 7, no. 1 (Jan. 1981)-v. 41, no. 10 (June 1996).
 550 ___ |a Published for: Euromicro.
 590 ___ |a SERBIB/SERLOC merged record
 650 _0 |a Microprocessors |v Periodicals.
 650 _0 |a Microprogramming |v Periodicals.
 650 _7 |a Microprocessors. |2 fast |0 (OCoLC)fst01020008
 650 _7 |a Microprogramming. |2 fast |0 (OCoLC)fst01020033
 650 _7 |a MICROPROGRAMMING. |2 unbist
 650 _7 |a DATA PROCESSING. |2 unbist
 655 _0 |a Electronic journals.
 655 _7 |a Periodicals. |2 fast |0 (OCoLC)fst01411641
 710 2_ |a Euromicro.
 776 08 |i Online version: |t Microprocessing and microprogramming (Online) |w (DLC)
 2005233216 |w (OCoLC)41227882
 776 08 |i Online version: |t Microprocessing and microprogramming |w (OCoLC)564648579
 776 08 |i Online version: |t Microprocessing and microprogramming |w (OCoLC)631723842
 780 00 |t Euromicro journal |x 0167-3858 |w (DLC) 78644982 |w (OCoLC)4109280
 785 00 |t Journal of systems architecture |x 1383-7621 |w (DLC) 96658139 |w (OCoLC)35360253
 850 ___ |a CLSU |a CSt |a CStclU |a CU |a CU-A |a CU-
 SB |a CaAEU |a CaOWtU |a DGW |a DLC |a GAT |a IU |a IaAS |a InLP |a MdU |a MiDW |a MiU |a
 MoU |a MsU |a NN |a NNE |a NcD |a NcU |a NjR |a OCU |a OkU |a PBL |a PPiC |a PPiU |a TxDaM
 |a TxU |a ViBibV |a WaU
 859 41 |u http://www.sciencedirect.com/science/journal/01656074
 890 ___ |a Microprocessing and microprogramming. |i 86-641015
 906 ___ |a 7 |b cbc |c serials |d 3 |e ncip |f 19 |g n-oclcserc
 920 ___ |a Keep 1
 985 ___ |e eserial 200406
 991 ___ |b c-GenColl |h QA76.5 |i .E89a |w SERIALS
 991 ___ |b c-GenColl |h QA76.5 |i .E89a |p 00034620362 |t Copy 1 |v 13-14 * 1984 : Feb.-Dec. |w CCF
 991 ___ |b c-GenColl |h QA76.5 |i .E89a |p 00034620404 |t Copy 1 |v 15-16 * 1985 |w CCF
 991 ___ |b c-GenColl |h QA76.5 |i .E89a |p 00034620441 |t Copy 1 |v 17-18 * 1986 |w CCF
 991 ___ |b c-GenColl |h QA76.5 |i .E89a |p 00034620489 |t Copy 1 |v 19 * 1987 : Jan. |w CCF
 991 ___ |b c-GenColl |h QA76.5 |i .E89a |p 00034620490 |t Copy 1 |v 31-32 * 1991 : Apr.-Aug. |w CCF
 991 ___ |b c-GenColl |h QA76.5 |i .E89a |p 00034620532 |t Copy 1 |v 33-34 * 1991 -92: Sept.-Feb.
 |w CCF

991 ___ **|b** c-GenColl **|h** QA76.5 **|i** .E89a **|p** 0003462057A **|t** Copy 1 **|v** 35 * 1992 : Sept. **|w** CCF
991 ___ **|b** c-GenColl **|h** QA76.5 **|i** .E89a **|p** 00034620611 **|t** Copy 1 **|v** 36-37 * 1992 -93: Nov.-Jan. **|w** CCF
991 ___ **|b** c-GenColl **|h** QA76.5 **|i** .E89a **|p** 00034620659 **|t** Copy 1 **|v** 38-39 * 1993 : Sept.-Dec. **|w** CCF
991 ___ **|b** c-GenColl **|h** QA76.5 **|i** .E89a **|p** 00034620696 **|t** Copy 1 **|v** 40 * 1994 **|w** CCF
991 ___ **|b** c-GenColl **|h** QA76.5 **|i** .E89a **|p** 00016229374 **|t** Copy 1 **|v** 41 APR-NOV * 1995 **|w** CCF
991 ___ **|b** c-GenColl **|h** QA76.5 **|i** .E89a **|p** 00034620325 **|t** Copy 1 **|v** 7:1 * 1981 : Jan. **|w** CCF
992 ___ **|b** SER **|w** SERLOC

[Request this Item](#)[LC Find It](#)

Item Availability



CALL NUMBER [QA76.5 .E89a](#)

Request in Jefferson or Adams Building Reading Rooms

Status Not Charged

CALL NUMBER [QA76.5 .E89a](#)
Copy 1

Request in Jefferson or Adams Building Reading Rooms

Status Not Charged

Older receipts 13-14 * 1984 : Feb.-Dec.
 15-16 * 1985
 17-18 * 1986
 19 * 1987 : Jan.
 31-32 * 1991 : Apr.-Aug.
 33-34 * 1991 -92: Sept.-Feb.
 35 * 1992 : Sept.
 36-37 * 1992 -93: Nov.-Jan.
 38-39 * 1993 : Sept.-Dec.
 40 * 1994
 41 APR-NOV * 1995
 7:1 * 1981 : Jan.

APPENDIX 1005C

Engr.

Microprocessing and Microprogramming

ABF-3104

VOLUME 38, NUMBERS 1-5
SEPTEMBER 1993

ISSN 0165-6074
MMICDT 38 (1-5) (1993) 1-862

The EUROMICRO Journal

QA
76
.5
E89
C5

COMPLETE VOLUME:
PROCEEDINGS EUROMICRO 93
OPEN SYSTEM DESIGN:
HARDWARE, SOFTWARE AND
APPLICATIONS
Barcelona, September 6-9, 1993

NORTH-HOLLAND

Published for EUROMICRO by NORTH-HOLLAND
(a division of Elsevier Science Publishers B.V.)



EUROMICRO 93

Nineteenth EUROMICRO conference

Barcelona

September 6-9, 1993

Open System Design:
Hardware, Software and Applications

Programme

Welcome to Barcelona

The nineteenth EUROMICRO Conference will be held from September 6th to 9th at the Polytechnic University of Catalunya in Barcelona.

The Conference main theme of this year is 'Open System Design: Hardware, Software and Application'. The purpose of the Conference is to bring together people from Business, Industry, Research, Government and Academia who are interested in all problems related to microelectronics and computing. EUROMICRO 93 aims to cater for a 'state of the art' coverage of current developments and future trends.

The Polytechnic University of Catalunya and the organizers extend a warm welcome to the EUROMICRO 93 Conference and its participants.

Programme

The EUROMICRO Conference is a major annual event around which a collection of related events congregate. EUROMICRO 93 will include a mainstream programme based on Keynote Sessions, Scientific Paper Sessions organized in four parallel tracks and Short Note sessions. Several other activities are organized such as a Tutorial and several Social Events.

Euromicro 93 has attracted a large number of submitted papers which were carefully reviewed - four referees each - by an international panel of 166 referees. The Programme Committee selected 94 papers arranged in 33 sessions.

Keynotes

Each day of the conference an outstanding keynote speaker will spotlight the theme of the Conference from a different point of view. Keynote addresses will be given by:

M. Silva of University of Zaragoza (E) on "Formal modelling and analysis of architectures using Petri Nets"

L.A. Zadeh of University of California (USA) on "Fuzzy Logic, Neural Networks and Soft Computing"

J.A.B. Fortes of Purdue University (USA) on "Matching algorithms and architectures"

Scientific and Technical Programme

The Scientific and Technical Programme has been organized by Programme Subcommittees.

- Software Engineering
Coordinator: P. Milligan (UK)
- System Design
Coordinator: A. Pawlak (D)
- Computer Communications and Networks

Coordinator: K. Tarnay (H)

- Parallel Processing
Coordinator: S. Winter (UK)
- Components and VLSI Design
Coordinator: A. Antola (I)

Short Notes

In parallel there is a Programme of Short Note sessions with contributions providing topical reports of recent works in all fields relevant to the Conference. The detailed Short Note Programme will be available at the Conference.

Tutorial

Monday, September 6th, features the tutorial Applying Concurrent Engineering in a Software Process by Marko Heikkinen (SF).

Social programme

During the EUROMICRO 93 Conference the following events are scheduled:

On Monday, September 6th., a Get-Together Party will be offered on the University campus (building TG on the map) from 17:00 - 18:30.

On Tuesday, September 7th., a reception by the Mayor of Barcelona will be held at 19:00 at the 'Palau de Pedralbes', a palace located near the campus. A classical music quartet will give a performance and during the reception a cocktail will be offered.

On Wednesday, September 8th. at 21:00, a Banquet will be organized in the 'Poble Espanyol', a charming place near the Olympic area, where you can visit some samples of architectural works from different places in Spain.



La Sagrada Família



Microprocessing and Microprogramming

The Euromicro Journal

Editors-in-chief

Mariagiovanna Sami
Politecnico Istituto di Elettronica
Piazza L. da Vinci, 32
I-20133 Milano, Italy
Tel.: (2) 2399 3516; telefax: (2) 2399 3411;
email: sami@ipmef1.polimi.it

Lutz Richter
Univ. Zürich-Irchel
Institut für Informatik
Winterthurerstrasse 190
CH-8057 Zürich, Switzerland
Tel.: (1) 257 4330; telefax: (1) 363 0035;
email: richter@ifi.unizh.ch

Editorial board

Software engineering

Gerhard Chroust
Johannes Kepler Universität Linz
Institut für Systemwissenschaften
Abt. für Systemtechnik und Automation
Altenbergerstraße 69
A-4040 Linz, Austria
Tel.: +43 (732) 2468-866
Telefax: +43 (732) 2468-822

Software and system architectures

Peter Milligan
Queen's University of Belfast
Department of Computer Science
University Road
Belfast, BT7 1NN,
United Kingdom
Tel.: +44 (232) 245 133 ext. 3293/3229
Telefax: +44 (232) 331 232
email: p.milligan@qub.ac.uk

Hardware architectures, communications and networks, embedded and real-time systems

N. Scarabottolo
Politecnico di Milano
Istituto di Elettronica
Piazza L. da Vinci, 32
I-20133 Milano, Italy
Tel.: +39 (2) 2399 3512
Telefax: +39 (2) 2399 3411

Distributed systems

Ferenc Vajda
KFKI Research Inst. for Measurement and Computing Technique
Dept. MSZKI, P.O. Box 49
1525 Budapest 114, Hungary
Tel.: +36 (1) 169 5532; telefax: +36 (1) 169 5532
email: vajda@iif.kfki.hu

Parallel architectures

Stephen Winter
Polytechnic of Central London
115 New Cavendish Street
London, W1M 8JS,
United Kingdom
Tel.: +44 (71) 911 5099
Telefax: +44 (71) 911 5089

A list of EUROMICRO Directors and Correspondents is given in the back of this issue.

© 1993, Elsevier Science Publishers B.V. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publisher, Elsevier Science Publishers B.V. Copyright and Permissions Department, P.O. Box 521, 1000 AM Amsterdam, Netherlands.

Submission of an article for publication implies the transfer of the copyright from the author(s) to the publisher and entails the author's irrevocable and exclusive authorization of the publisher to collect any sums or considerations for copying or reproduction payable by third parties (as mentioned in article 17 paragraph 2 of the Dutch Copyright Act of 1912 and in the Royal Decree of June 20, 1974 (S. 351) pursuant to article 16b of the Dutch Copyright Act of 1912) and/or to act in or out of Court in connection therewith.

Special regulations for readers in the USA—This journal has been registered with the Copyright Clearance Center, Inc. Consent is given for copying of articles for personal or internal use, or for the personal use of specific clients. This consent is given on the condition that the copier pays through the Center the per copy fee stated in the code on the first page of each article for copying beyond that permitted by Sections 107 or 108 of the US Copyright Law. The appropriate fee should be forwarded with a copy of the first page of the article to the Copyright Clearance Center, Inc., 27 Congress Street, Salem, MA 01970, USA. If no

code appears in an article, the author has not given broad consent to copy and permission to copy must be obtained directly from the author. All articles published prior to 1981 may be copied for a per-copy fee of US \$ 2.25, also payable through the Center. This consent does not extend to other kinds of copying such as for general distribution, resale, advertising and promotion purposes, or for creating new collective works. Special written permission must be obtained from the publisher for such copying.

Special regulations for authors—Upon acceptance of an article by the journal, the author(s) will be asked to transfer copyright of the article to the publisher. This transfer will ensure the widest possible dissemination of information under the US Copyright Law.

This journal is printed on acid-free paper.

Published 15 times a year

Printed in The Netherlands

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

Although all advertising material is expected to conform to ethical standards, inclusion in this publication does not constitute a guarantee or endorsement of the quality or value of such product or of the claims made of it by its manufacturer.

OPEN SYSTEM DESIGN: HARDWARE, SOFTWARE AND APPLICATIONS

Welcome to Barcelona, the seat of the Nineteenth EUROMICRO Symposium.

Barcelona, founded by the Phoenicians, facing the Mediterranean Sea, has been a city of new blood to the vitality of its people.

Nineteenth EUROMICRO Symposium on Microprocessing and Microprogramming (EUROMICRO 93)

Barcelona, September 6-9, 1993

edited by

Ferenc Vajda

and

Benny Graff Mortensen

EUROMICRO 93 is the nineteenth Euromicro Conference. This annual event is an European leading forum for people from business, industry, research, government and academia who are interested in all problems related to microelectronics and computing. The Conference is attended by people from most countries in Europe, the United

States, Africa and the Far East.

This year's Conference features five tracks in the topics of System Design, Computer Communications and Networks, Components and VLSI Design, Software Engineering and Parallel Processing. The program includes three keynote sessions with invited papers, scientific sessions in the five tracks of the Conference and short-note sessions reporting on significant results and applications of recent works. In addition, the first day of the Conference is dedicated to tutorials.

The EUROMICRO 93 Conference is held at the new part of the campus of the Universitat Politècnica de Catalunya in Barcelona. The campus is located in a quiet area of the city with excellent transport facilities. The participants will find there the right atmosphere for an exchange of ideas.

We are most grateful for the support of the Universitat Politècnica de Catalunya and the Ministry of Education of Spain. We also thank the Euromicro organization for their assistance. We are indebted to Chiquita Salgue-Muñoz from the European Office for her continuous support. Finally, we wish to thank all the delegates participating in the Conference and we extend to them a warm welcome to Barcelona.

*Manu Valls
Joan Chacón
Antonio Chacón*



1993

NORTH-HOLLAND — AMSTERDAM • LONDON • NEW YORK • TOKYO

Table of Contents

Chairmen's Introduction	vii
Program Chairman's Introduction	ix
Reviewers	xi
Session Chairpersons	xiii
K1: Petri Nets applied to the Modelling and Analysis of Computer Architecture Problems (Invited paper) M. Silva and J.M. Colom (E)	1
K2: Fuzzy Logic, Neural Networks and Soft Computing (Invited paper) L.F. Zadeh (USA) (<i>abstract</i>)	13
K3: Matching Algorithms and Architectures (Invited paper) J.A.B. Fortes (USA) (<i>abstract</i>)	15
A1: Interconnection Networks Chairperson: K. Tarnay (H)	17
Hardware Features of the Static Communication Network of a Parallel Architecture V. Néri, J.-L. Béchenec, F. Cappello, D. Etienne (F)	19
Fault Tolerant Multiple-bus Interconnection Networks J. Sosnowski (P)	25
Experimental Evaluation of Mad Postman Bidimensional Routing Networks C. Izu (E), R. Beivide (E), C. Jesshope (UK) and A. Arruabarrena (E)	33
A2: Processor Architecture I. Chairman: K. Waldschmidt (D)	43
Evaluating Transport Triggered Architectures for Scalar Applications H. Corporaal (NL)	45
MOVE32INT, a Sea of Gates Realization of a High Performance Transport Triggered Architecture H. Corporaal and P. van der Arend (NL)	53
Design of a VLIW Architecture Constructed from Standard RISC Chips: A Case Study of Hardware/Software Codesign P. Pfahler, C. Nagel, F.-J. Rammig, U. Kastens (D)	61

A3:	System Specification I. Chairman: W. Grass (D)	69
	X-Nets: a Visual Formalism for System Specification and Analysis S. Antoniazzi, A. Balboni and W. Fornaciari (I)	71
	Automated Support for the Formal Specification and Design of Real-Time Systems C. Ho-Stuart, H. Zedan and M. Fang (UK)	79
	SEAL. A Simple Language for Prototyping Action-event Specifications K. Brink, R. Huijsman and J. van Katwijk (NL)	87
B1:	Distributed Memory Management Chairman: P. Milligan (UK)	97
	The Memory Hierarchy of the CHESS Computer D. Lioupis, N. Kanellopoulos and M. Stefanidakis (GR)	99
	Some Practical Considerations for the Implementer of the SCI Network V. Habchi, U. Finger and C. O'Donnell (F)	109
	Conflict-free Access to Streams in Multiprocessor Systems M. Peiron, M. Valero, E. Ayguadé and T. Lang (E)	119
B2:	Processor Architecture II. Chairman: J. Cortadella (E)	131
	Evaluating the Cost of Conditional Branches on the Performance of Superscalar Machines E.S.T. Fernandes, F.M.B. Barbosa and D.M. Simpson (BR)	133
	Branch Optimization of the TOOBSE2 RISC-Processor and Classification M. Schäfers (D)	141
	Arco: A Cost-Effective and Flexible Hardware Maze Router J.S. Aude, E.P.L. Filho, M.F. Martins and S.B. Pinto (BR)	149
B3:	Network Management Chairman: J. Labarta (E)	161
	Protection and Routing Algorithms for Network Management C.V. Papadopoulos and A.H. el Zahni (GR)	163
	DIP: Distributed Diagnosis Protocol L.M. Silva and J.G. Silva (P)	171

	PLANNER 4: A Simulator to Design Packet Switching Data Network S. Allegra, M. Annunziata, F. Cesaroni and C. Lunadei (I)	179
C1:	Parallel Software Engineering Chairman: P. Kacsuk (H)	187
	Tools For Object-Oriented SPMD Programming B. Moisan, Y. Duthen and R. Caubet (F)	189
	Distributed Active Objects: A Methodological Proposal and Tool for Distributed Programming with Transputer Systems M. Capel, J.M. Troya and A. Palma (E)	197
	A Reusable Load Balancer for Parallel Search Problems W. Joosen, S. Bijnens and P. Verbaeten (B)	205
	Real-time Behaviour Monitoring for Multi-processor Systems J.P. Calvez and O. Pasquier (F)	213
C2:	VLSI Testing and Testability Chairman: I. Erényi (H)	221
	Fault Behaviour and Testability of Asynchronous CMOS Circuits H.T. Vierhaus, W. Meyer, U. Gläser and R. Camposano (D)	223
	FSM Fault Models Impact on Test Performances C. Bolchini and F. Fummi (I)	229
	Redundancy Analysis and Removal for VLSI ASIC's K. Ivinskis (D)	237
	Memory Testing in a Massively Parallel Machine Ch. Aktouf, Ch. Robach and G. Mazaré (F)	245
C3:	Advances in Object-Oriented Design Chairman: K. Klöckner (D)	253
	Active Object Oriented Databases in Control Applications P. Loborg, T. Risch, M. Sköld and A. Törne (S)	255
	Heterogeneous Schema Integration Method for Multidatabase System Y. Lee and S. Moon (ROK)	265
	Toward Automated Support for Object-Oriented Modelling J. Ayre, D. Glass, J.G. Hughes and I.R. McChesney (UK)	273

	Incremental Resource Estimation with Real-Time Feedback from Measurement M. Oivo (SF)	281
C4:	Local Area Networks Chairman: H. Rzehak (D)	291
	Concurrent Ring: Design and Evaluation of a New Token Ring LAN J.D. Sandoval, F.A. Herrera, A. Suárez and C. Sandoval (E)	293
	A Distributed Bandwidth Allocation Algorithm for Gbit/s LANs T. Antonakopoulos, S. Koutroubinas, J. Koutsonikos and V. Makios (GR)	301
	An ISDN Primary Rate Interface for Ethernet Access F. Videira and A. Casaca (P)	309
D1:	Parallel Algorithms Chairman: E. Zapata (E)	317
	Parallel WZ Factorization on Mesh Multiprocessors R. Asenjo, M. Ujaldón and E.L. Zapata (E)	319
	Image Component Labelling on Reconfigurable Processor Array M. Maresca, P. Baglietto and A. Giordano (I)	327
	Computing the Generalized Eigenvalues of Symmetric Positive Definite Pencils on Networks of Transputers D. Giménez, V. Hernández and A.M. Vidal (E)	335
D2:	ASIC Design Chairman: N. Scarabottolo (I)	343
	Efficient Synthesis of Distributed Vector Multipliers B. Fehér (H)	345
	Implementation of Given's Rotation Processors for DSP Real-Time Applications D.E. Metafas, E.P. Mariatos, S.S. Nikolaidis and C.E. Goutis (GR)	351
	On Computing the 2-D Discrete Cosine Transform Using Rotations H.C. Karathanasis (GR)	359
D3:	Data Base Systems Chairman: A.M. Tyrrell (UK)	367
	Visual Query Language for Object-Oriented Databases: OQD J.C. Kwak and S. Moon (ROK)	369

	Global Query Management in Heterogeneous Distributed Database Systems S. Kang and S. Moon (ROK)	377
	A Distributed Deadlock Detection Algorithm for the AND Model J.R.G. de Mendivil, C.F. Alastruey and J.R. Garitagoitia (E)	385
D4:	Logic Design Chairman: A. Jávör (H)	393
	A Study of the Resetability of Synchronous Sequential Circuits A. Lioy and M. Poncino (I)	395
	Relational Algebra as Formalism for Hardware Design A.J.W.M. ten Berg, C. Huijs and Th. Krol (NL)	403
	An Object-Oriented Model of Design Evolution C. Charlton, P. Leng and M. Rivers (UK)	411
E1:	Parallelizing Compiler Techniques Chairman: R. McConnell (UK)	419
	Exact and Efficient Advanced Loop Interchange C-S. Yang, C-C. Chang, J-M. Yang, T-C. Huang and K.C. Huang (TW)	421
	Resource-Constrained Pipelining Based on Loop Transformations F. Sánchez and J. Cortadella (E)	429
	Optimal Data Dependence Chaining in Parallel Loops Z. Szczerbiński (P)	437
E2:	Microcontroller and Microprocessor Design A. Nunez (E)	445
	An Asynchronous Approach to the RISC Design of a Micro-Controller A. Costa, A. De Gloria, P. Faraboschi, G. Nateri and M. Olivieri (I)	447
	Application of the Novel Associative Programmable Array-structure Multi-Match-PLA in Synthesis of Decomposed Finite State Machines J. Kottsieper and K. Waldschmidt (D)	455
	OCEAN: A Hardware and Software Tool for Design of Future Smart Cards O. Caron, V. Cordonnier and G. Grimonprez (F)	467
E3:	Real-time Systems Chairman: J. Karjalainen (SF)	475

	Implementing and Checking Timing Constraints in Real-Time Programs K-J. Lin and K.B. Kenny (USA)	477
	On Guaranteeing Hard Real-Time Tasks C.E. Moron and H. Zedan (UK)	485
	Priority-Driven Concurrency Control Based on Data Conflict State in Distributed Real-Time Databases J. Kim and H. Shin (ROK)	491
E4:	VLSI Simulation Chairman: F. Distanto (I)	501
	SHC-SLX: A Levelized Compiled, Event Driven Interpreted VLSI Simulator L. Carro, C.A. Marcon and A.A. Suzim (BR)	503
	Timing Analysis for DCFL/SDCFL VLSI Circuits L. Gómez, A. Hernández and A. Nunez (E)	511
	Sequential and Parallel Strategies for the Demand-Driven Simulation of Logic Circuits P.E. Dunne, C.J.J. Gittings and P.H. Leng (UK)	519
F1:	Application-Driven Architectures Chairman: S. Winter (UK)	527
	A Processor Array Module for Distributed, Massively Parallel, Embedded Computing L. Bengtsson, K. Nilsson and B. Svensson (S)	529
	Design of an Extended Transputer Processor Farm System C.W. Yung and E.K.N. Yung (HK)	539
	A Parallel Architecture for the Color Doppler Flow Technique in Ultrasound Imaging A. Costa, A. De Gloria, P. Faraboschi and M. Olivieri (I)	545
	An Empirical Study of Automatic Restructuring of Vision Programs for SIMD Machines L.A. Barragán and A. Roy (E)	553
F2:	The FORMAT Project (ESPRIT 6028) Chairman: J. Goicolea (E)	561
	A Design Methodology for the Correct Specification of VLSI Systems C. Bolchini, M. Bombana, P. Cavalloro, C. Costi, F. Fummi and G. Zaza (I)	563

	VHDL/S - Integrating Statecharts, Timing Diagrams, and VHDL J. Helbig, R. Schlör, W. Damm, G. Döhmen and P. Kelb (D)	571
	Introducing Structure into Behavioural Descriptions Obtained from Timing Diagram Specifications W.D. Tiedemann, S. Lenk, C. Grobe and W. Grass (D)	581
	VHDL Generation from a Timed Extension of the Formal Description Technique LOTOS within the FORMAT Project C.D. Kloos, T. de Miguel Moro, T.R. Valladares, G.R. Filho and A.M. López (E)	589
	Analysis Tools Applied to VHDL S. Olcoz and J.M. Colom (E)	597
F3:	From Requirements to Implementation Chairman: M.J. Taylor (UK)	605
	Real-Time Specification Using Petri Nets K. Sacha (PL)	607
	IDFG: An Interactive Applications Specification Model with Phenomenological Properties A. Kameas, S. Papadimitriou, P. Pintelas and G. Pavlides (GR)	615
	Three Approaches to Interprocedural Dynamic Slicing M. Kamkar, P. Fritzson and N. Shahmehri (S)	625
	A Software System for the Generation of Test Data for Ada Programs M.J. Gallagher and V.L. Narasimhan (AUS)	637
F4:	Issues in Distributed Systems Chairman: K.P. Judmann (A)	645
	Concurrency Control Algorithms for Real-Time Systems H. Nakazato and K-J. Lin (USA)	647
	Analysis and Simulation of Six Bus Arbitration Protocols V.L. Narasimhan and S. Price-White (AUS)	655
	Design of TMS320C40 Signal Processors and Programmable Logic Based Prototyping Environment of Real-time Machine Vision Architectures K. Rautiola, K. Pehkonen, L. Stähle and P. Jokitalo (SF)	663
	Modelling and Design of the Multimedia Subsystem of a Distributed Authoring Environment A. Kameas, S. Papadimitriou and P. Pintelas (GR)	669

G1:	Neural Networks G. Kock (D)	679
	Mapping and Parallel Simulation of Synchronous Neural Networks on Multiprocessors T. Reski (D)	681
	Neural Accelerator for Parallelization of Back-Propagation Algorithm E. Franzi (CH)	689
	Removal of Catastrophic Noise in Hetero-Associative Training Samples E. Tuv and A.N. Refenes (UK)	697
G2:	Algorithms and Tools for VLSI Design Chairman: L. Jóźwiak (NL)	705
	An Empirical Study of Genetic Operators in Genetic Algorithms Xin Yao (AUS)	707
	Automatic Test Pattern Generation can be Solved as a Constraint Satisfaction Problem K. Tilly, L. Surján and G. Román (H)	715
	A Behavioral Level Tool for Design and Verification of Parallel Architectures: PADS F. Distanto and P. Toccaceli (I)	723
G3:	Protocols and Performance Analysis Chairman: G. Ciccarella (I)	731
	Providing Circuit Service over a High Speed Deflection Network E. Pagani and G.P. Rossi (I)	733
	Dynamic Protocol Configuration for Multimedia Networks R. Stainov (D)	741
	Performance Analysis of Synchronous Multibuffered Packet-Switching Networks A. Tentov and A. Grnarov (MC)	749
G4:	RT-level Synthesis Chairman: F. Tirado (E)	757
	Automata Decomposition and Synthesis with PLAM S. Baranov and L. Bregman (IL)	759

	Synthesis for Testability by Synthesis Controlling C. Nagel (D)	767
	GLASS: a Graph-Theoretical Approach for Global Binding R.M. Badia and J. Cortadella (E)	775
H1:	Fault Tolerance Chairman: K.E. Grosspietsch (D)	783
	The Design of Time-critical Conversations A.M. Tyrrell and G.F. Carpenter (UK)	785
	A Fault-Tolerant Server on MACH S. Arévalo, J. Carretero, J.L. Castellanos and F. Barco (E)	793
H2:	Decision Making Under Uncertainty Chairman: A. Rea (UK)	801
	Pen-based Recognizing of Handprinted Characters B. Klauer and K. Waldschmidt (D)	803
	N Version Programming with Majority Voting Decision: Dependability Modelling and Evaluation K. Goševa-Popstojanova and A. Grnarov (MC)	811
H3:	FDT Tools Chairman: H. Schumny (D)	819
	PROCONSUL, a Tool for Computer-aided Protocol Engineering K. Tarnay, Gy. Csopaki, M. Törő, M.D. Ta, Zs.B. Harangozó and J. Miskolczi (H)	821
	Test Strategy for Indeterministic Protocols M. Törő (H)	827
H4:	System Specification II. Chairman: D. Sanders (UK)	833
	TL: A System Specification System L.P.M. Benders and M.P.J. Stevens (NL)	835
	Connectionist Model Description: a Case Study G. Kock and N.B. Šerbedžija (D)	843
	Session Chairpersons' addresses	851
	Author index	855
	Subject index	859

DIP : Distributed Diagnosis Protocol

Luís Moura Silva and João Gabriel Silva

Laboratório de Informática e Sistemas
Universidade de Coimbra
Urb. Boavista, Lt. 1-1
3000 Coimbra - Portugal
Fax : +351.39.701266
e-mail : lams@uc.pt

Abstract

This paper presents the DIP - a Diagnosis Protocol to check the availability of processors in a distributed system. Each node maintains a Network Status Table, and failure detection is achieved by alive messages. The overhead of such messages is minimal and the main virtue of the protocol is the lower diagnosis latency when a failure is detected. Since the unavailability of a node can be the result of a processor crash or a network partition, the DIP protocol is also able to cope with partitions. The protocol can diagnose fail-stop and omission failures. Moreover, it is robust against out-of-order messages, duplicates and false-alarms.

Keywords : *Distributed Diagnosis, Fault Tolerance, Membership Protocols, Network Monitoring.*

1. Introduction

One of the requirements of a fault-tolerant distributed system is the failure detection capability of remote sites. Recovery and reconfiguration are other desirable properties. Since the paper of Preparata, Metze and Chien [1] there were further theoretical developments [2] in the area of distributed diagnosis. As a result, several PMC-based diagnosis algorithms have been suggested [3-9]. Recently, Bianchini *et al.* [10,11] have conducted the first implementations of system diagnosis theory in a real distributed network environment. Nevertheless, all those PMC-based algorithms have serious drawbacks:

1) when the number of node failures exceeds the connectivity of the system digraph, those algorithms simply yield false diagnostic results. 2) The PMC-model is only valid for permanent processor failures and it assumes that a node cannot fail and restart in the period between two tests. 3) Most of the algorithms require a high number of diagnostic messages and, as a result, the execution time may exceed reasonable values.

All these shortcomings make the algorithms unsuited to local area networks. Moreover, if one considers that processors may exhibit byzantine faultiness [12], PMC-based algorithms may identify

none of the faulty processors [13]. Another class of diagnosis algorithms which identifies byzantine faults has been suggested [13-15]. These algorithms are able to diagnose intermittent and malicious faults. However, the diagnosis of byzantine faults becomes extremely difficult and all the proposed solutions present a great message overhead. As practitioners, we are not willing to pay such price and we simply assume a more restricted failure model: the crash failure model. In this model it is assumed that each node can only fail in "clean" well defined ways. During its functioning the node behaves correctly and if an internal failure occurs it halts automatically. Such nodes are called fail-silent or fail-stop. A lot of algorithms [18-27] have also assumed this failure model but no one seems to perfectly suit our goals and assumptions.

2. System Model and Assumptions

The system consists of a set of N stations interconnected by a LAN with broadcast capability. Each station has access to *stable storage* which can be used to store information that must survive to failures. In perfect conditions each processor can communicate with any other processor in the network. We assume that the relation "can-communicate" between nodes is symmetric and

transitive. Two kinds of failures are considered: processor failures and communication link failures. Processor failures are assumed to be clean. When a node fails it simply stops running and does not send any arbitrary or malicious messages. Thus, the universe of failures includes stop and omission failures and no assumption is made on the number of allowable processor failures.

Network failures can be classified in hard and soft failures [16]. An hard failure is characterized by the incapability of the network to deliver any bandwidth at all. Soft failures are characterized by performance degradations or deviations from the expected behaviour. In networks, such as the Ethernet [17], which do not provide any hardware level acknowledgment, the other nodes of a network cannot distinguish an isolated node from a crashed one, since in both cases the symptom is the same - the loss of communication with the node. When an active node is not available to any other node, it means that the network was partitioned (hard failure) and both nodes are in different partitions. However, crashed nodes can recover and partitions can be reconnected. For the sake of generality let us consider the existence of two main events : crashes and restarts. Each available node is marked as UP while an unavailable node is classified as DOWN. Each processor will have its own view of the network which must be consistent with that of other nodes belonging to the same partition.

We assume that the message delivery is not reliable since the network is subjected to soft failures. Messages can be lost, delayed or delivered out-of-order. However, a network soft failure cannot produce any malicious message.

We consider that there exists a maximum message delay ∂_{max} , which reflects the expected maximum transmission time of a single message between two nodes, including the processing delay. The minimum message delay is represented by ∂_{min} . Every node should have a physical clock in order to implement timers but we do not require clocks to be synchronized.

3. Design Guidelines

The DIP protocol should be able to perform three functions :

- (1) To detect node crashes.
- (2) To notify all the network nodes of the crash of a processor (Node Down event).

- (3) and also to notify all the nodes of the restart of a crashed node (Node Up event).

Each node maintains a table with the status of the other nodes, called the *Network Status Table* (NST). Each entry of the NST has three fields : the host name, its status and the *Incarnation Number* (IN). The IN number is helpful to distinguish duplicates and out-of-order messages, as will be explained in §4.4.

Since failures are the exception rather than the rule, the first imperative is to reduce the overhead during normal operation. The protocol must assure a minimal number of messages during steady-state operation. Another important goal is to reduce the diagnosis latency to a minimal value, i.e., the protocol must provide timely failure detection. This is a paramount figure of merit to allow an effective recovery/reconfiguration of distributed applications. When a node failure occurs all the other nodes must be acquainted with that as quickly as possible. As failures and restarts are asynchronous events, the protocol should be robust against their occurrence during its execution. Other important issue is the handling of network partitions. The other desired goals are efficiency, simplicity and correctness of the protocol.

4. Design Details

4.1 The Detection of Failures

As a fail-silent processor only stops when an internal fault occurs the failure can be simply detected by background *I_AM_ALIVE* messages. The network nodes are organized on a virtual ring and all the sites are numbered from 0 up to N-1, along the path of the virtual ring. It should be noted that the ring does not reflect the physical organization of the network. The order of the nodes in the ring follows the order of the Status Table, which is unique to all the processors in the network.

Each processor P_i sends periodically an *I_AM_ALIVE* message to the next neighbour $P_{((i+1) \bmod N)}$ while it is checking for the messages sent by the previous processor $P_{((i-1) \bmod N)}$. Each processor has two independent processes to perform that job.

- 1) The *Heartbeat* process is periodically awakened by a timer interrupt in order to send a message to the next neighbour. This timer is represented by T_H and is set to α .

2) The detection of failures is done by the Watchdog process, which is listening to the messages sent by the previous neighbour. Every time it receives an `I_AM_ALIVE` message it triggers a timer T_W with $((\Omega+1).\alpha + \partial_{\max} - \partial_{\min})$, where Ω is the number of omission failures or lost messages that can be tolerated. If the timer expires, this means that the previous node has crashed and the Node Down Protocol is started for that node.

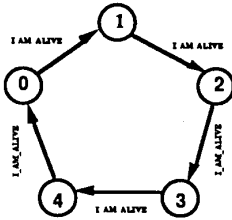


Figure 1 : Virtual Ring

4.2 Node Down Protocol

In order to reduce the diagnosis latency, the node should notify all the others immediately by a broadcast message to the network, announcing the `Node_Down` event of its neighbour.

Take as an example the network of figure 2. One can notice that node #2 has crashed and its tester (node #3) will detect that when its timer T_W expires. The status of node #2 is marked as DOWN in its Network Status Table, and it will send a broadcast message to the network, which includes the number of the node and the `NST(2).IN` : `NODE_DOWN(2,1)`.

In a meantime, a timer T_{ack} will be set to $2.\partial_{\max}$. The node will inspect its status table to find the available nodes and waits for the acknowledge of every UP node. On the other hand, when a node receives a `NODE_DOWN` broadcast it replies with a `ND_Ack` message to node #3. It should be noted that each acknowledge is also a broadcast message. This may be useful if a particular node does not receive the broadcast message of #3 it may be informed of that by one of the acknowledge messages. When node #3 receives all acknowledges this phase is finished and the protocol returns to the steady state. However, if the timer T_{ack} expires and some reply is missing, the message `NODE_DOWN(2,1)` is retransmitted again (until a maximum number of retries is reached).

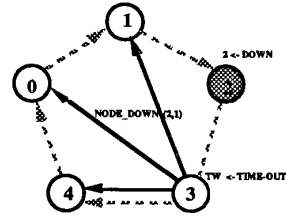


Figure 2 : Detection of a Failure

It may happen that the unavailability of node #2 is not the crash of a single processor, but instead the network may be partitioned. In this case, some nodes may not answer to the message and should also be classified as DOWN. In fact, if after the retransmission, node #3 does not receive any reply then it will start again this protocol announcing the unavailability of those nodes which remained in the other partition of the network.

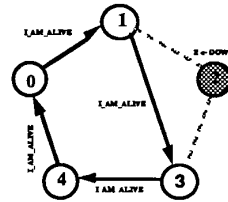


Figure 3 : Reconfiguration

After the notification phase of a node crash, the testing structure has to be adapted to a new configuration. Each node that had been notified that its tester neighbour has crashed must identify the next available node, by searching the Status Table (see example of node #1 in figure 3). The detector node (#3) will also have a new tested node (#1).

4.3 Node Up Protocol

When a node rejoins the network it must tell the others that it is back in operation and must also determine which nodes are UP and reachable. Therefore, when a node recovers it reads the NST from stable storage to main memory. The status of all the other nodes are set to DOWN. Then it increments its IN value, which is also stored in stable storage, and sets to zero all the counters of the other nodes. Finally, it broadcasts a `NODE_UP` message to notify all the active nodes in the

network. A timer T_{up} is activated. This message must be acknowledged by the other available nodes. The replies will be used by the restarted node to achieve its new network view. Furthermore, to guarantee a distributed consistency, node #2 must verify whether all the nodes it can reach share the same NST.

Let us consider the previous scenario : node #2 is DOWN and its IN is equal to 1. Now, suppose that it recovers and rejoins the network. First of all, its IN counter will be incremented, and it will send a `NODE_UP` broadcast message to the network, which includes the new IN (see figure 4).

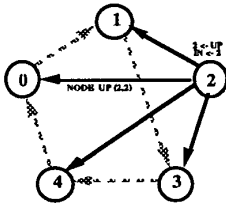


Figure 4 : Node #2 Rejoins the Network

When each of the other nodes receive the `NODE_UP(2,2)` message they check the 2nd entry of its NST, and the status of node #2 is marked as UP. Then, they send a `NU_Ack` message to node #2, which contains their own INs. Each node also appends a list of its DOWN nodes into that message, including the corresponding INs. When the timer T_{up} expires, node #2 will have a complete information about the network status. Likewise, to assure a distributed agreement, node #2 must check the information it had received from the `NU_Ack` messages. The DOWN lists which were piggybacked into the acknowledges must be all consistent. Otherwise, node #2 will have to broadcast the result of the comparison to force a distributed consensus.

4.4 Protocol Integration

As the two sub-protocols may execute simultaneously they must interact with each other. Let us consider the following situation : node #2 has crashed ; node #3 detects its failure and sends a `NODE_DOWN` broadcast message to the network. However node #2 is DOWN for only a short time, after that it comes up again and rejoins the network by sending a `NODE_UP` message. Yet, an

inconsistent scenario may occur: Node #0 receives the sequence (`NODE_DOWN`, `NODE_UP`) while node #1 receives the (`NODE_UP`, `NODE_DOWN`) sequence. Fortunately, the *Incarnation Number* can solve this inconsistency. Every message is tagged with the IN and each receptive node will compare the incoming IN with the value stored in its NST. When a host receives a `NODE_UP` message from node #K and the incoming IN exceeds $NST(K).IN$, the table is automatically updated. If the incoming IN is similar it is recognized as a duplicate message and no action is performed. In either cases a reply (`NU_Ack`) is sent. When a node receives a `NODE_DOWN` message from node #X notifying the crash of its neighbour (node #Y) it checks in its table the status and the current IN of node #Y. If #Y is UP and the incoming IN exceeds the $NST(Y).IN$, then #Y is marked as DOWN and the local IN is updated. In normal situations, the incoming IN is similar to the local value. In this case, only the status of node #Y is updated. On the other hand, if #Y is already DOWN the $NST(Y).IN$ will only be updated if the incoming IN exceeds the local IN. In either case the receiving node will reply to each message with its own IN.

Taking the previous example, let us assume that node #2 was running with $IN=1$. By the time it had crashed, node #3 would have sent a `NODE_DOWN` message announcing this event (with $IN=1$). When the node recovers, it would increment IN to 2.

Node #0 would receive the messages in the correct order. However, when node #1 received the message `NODE_UP(2,2)` it would realize that node #2 had crashed and came up again without being aware of that. It would set $NST(2).IN$ to 2 and send a `NU_Ack` message to node #2, which would include its own IN ($NST(1).IN$).

After having received the `NODE_DOWN(2,1)` message it would conclude that the message was out-of-date since the incoming IN was less than the stored one. In spite of that, a reply message (`ND_Ack`) would be sent to node #3.

4.5 False Alarms

A *false alarm* occurs when an active node receives a `NODE_DOWN` broadcast message notifying its own failure. It may happen if the number of lost messages exceeds the tolerated level. A network soft failure can be the cause of that. In this case, the node will react to that "injury" by broadcasting a

I'M NOT DOWN message to the network with its current IN. When the other nodes receive such message they undo the action taken by the reception of the *false alarm* message. This message is only taken into account if the incoming IN is equal to the stored value.

4.6 Coping with Network Partitions

Network partitioning is an event which causes the network to divide into two or more sub-networks. The worst case is when a single node is isolated from the rest of the network due to a crash of its communication processor. Partitioning should be detected during the execution of the Node Down Protocol.

To simplify the explanation let us consider the network represented in figure 5. Due to a communication link breakdown or to some other network component (e.g. gateway, bridge, router...) failure, the network was split into two sub-networks. The detection of this partitioning begins with the detection of two single node failures. Node #2 and node #0 will detect the crashes of node #1 and node #4, respectively. Each of the detectors will send a NODE_DOWN broadcast and then wait for the acknowledge messages. As Node #2 will note the lack of response from node #0 it will repeat the Down protocol for node #0. On the other hand, this node only receives the ND_Ack from node #1. Thus, it will broadcast to its sub-network the failure of node #2 and #3. Only node #1 will hear these messages.

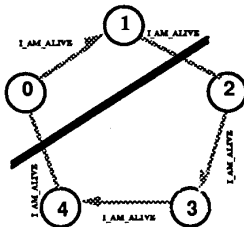


Figure 5 : Network Partitioning

The final result of this phase is represented in figure 6. Two partitions have been identified (P1 and P2) and the main graph of test has been divided into two independent graphs. To the nodes of partition P1, the nodes belonging to P2 are marked as DOWN, and vice-versa. The protocol guarantees that each time a network partitioning is detected this is assumed at

both sides of the "wall" between partitions.

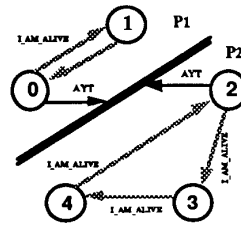


Figure 6 : Partitioning Reconfiguration

Furthermore, it is necessary to assure the detection of the possible merge of these partitions. This job is assigned to both *detector* nodes (#0 and #2). Each one will resume a *dummy* process that periodically sends a AYT² broadcast message to the network, which contains the identification of the sender and its IN. A rule is established to the nodes of the same partition : they do not answer to those messages. For instance, in partition P1 node #0 sends periodically an AYT broadcast message. Since node #0 is UP for node #1, it will not reply.

Those messages may pass through the partition "wall", and can be heard by the other nodes only when the two sub-networks get merged. Each node that receives an AYT message from a sender judged as DOWN, will acknowledge it with a YIA³ message. This reply contains the node identification, its current IN and is also broadcasted. A YIA works like a NODE_UP message with just one exception : if the incoming IN is similar to or exceeds the local IN, the message is accepted and the table is updated for the sender of the message. If the incoming IN is smaller, it is classified as obsolete. In both cases, a reply is sent to the network.

In the previous example, nodes #2 and #0 kept watching the other partition. They periodically send a AYT messages to the network. While the partition is maintained, those messages are not acknowledged. When the clusters get reconnected, node #1 and #0 hear the AYT (2) while nodes #2, #3 and #4 hear the AYT (0) message. They answer to those messages with a broadcast YIA and a special Node_Up protocol is started in each node. After that phase, the network recovers to the original configuration.

² AYT means "Are You There?"

³ YIA means "Yes, I Am !"

to $\log_2 N$. An experiment in a network of 60 processors with a version that uses an asymmetric forwarding scheme has shown that the overall diagnosis latency was about 90 seconds for one processor failure. The version that assumes the original model of forwarding of the PMC-model has an overall diagnosis latency of 450 seconds.

If one wants an efficient algorithm the latency should be drastically reduced and the number of messages minimized. The algorithms that assume the crash failure model are the most suitable and can be easily applied in real systems since they require a lower message count and are much more simple.

A centralized approach was chosen in the Publishing system [18]. In this case, a dedicated node, called the recorder, has a watchdog process for each processor of the network. If no messages are received in a time limit, the processor is considered to have crashed. This approach is simple but has a weak point in the recording node. Another centralized approach was chosen in [19], which uses an election algorithm to choose the master node.

At the other extreme, we found the detection mechanism of the TANDEM NonStop system [20], where each node tests every other node in the network. The DIP approach falls between the centralized and the fully distributed approach.

A practical algorithm can be found in [21]. It has some similarities with the DIP. However, that proposal has some shortcomings. First, it leaves the failure detection job to the application. Second, when a node fails, the number of diagnosis messages is $2N^2 - 6N + 4$ (where N is the number of nodes). This means that the network will be flooded with messages, and the diagnosis latency will suffer from that.

The algorithm presented by Walter [22] is similar to the DIP - it also organizes the network as a virtual ring of processors. But, one may notice that it is not robust against lost messages, duplicates, out-of-order messages and false alarms, since it assumes a reliable network. The critical message path of the diagnosis is 3, while the critical path of the DIP is just 1, which means that the diagnosis latency is at least 3 times higher than the latency of DIP. During steady-state operation the RSC protocol always requires the same number of messages: $P \cdot N$.

Another algorithm implemented in the SDD/1's RelNet [23] provides a systematic detection of failures and node recoveries. Nevertheless, that

algorithm is not robust against network partitions. Another set of algorithms has been proposed for synchronous distributed systems [24-27].

7. Implementation Features

The first release of the DIP was implemented on a set of SUN workstations which communicate through an Ethernet local area network. It was written in C using sockets UDP/IP. The protocol itself was running as a background process. The initialization module stores in stable storage the list of the processors in the network. This list is sorted by the Internet host address. As this number is unique to each processor it assures that all the hosts generate identical lists. All the crash and recovery events are stored in the *Crash Log* file, which can be inspected for failure analysis. A query interface is also available to users asking for the network status. The parameters chosen were the following: $\alpha = 1$ second and $\Omega = 2$.

The DIP provides a failure detection capability to be incorporated in a recovery mechanism, based on a distributed checkpointing algorithm presented elsewhere [28].

8. Conclusions and Future Work

We have presented a protocol for checking the availability of sites in local area networks. During steady-state operation it uses the minimum number of messages, while in case of failure it assures the lowest diagnosis latency. Its testing graph is adaptive which provides a complete diagnosability. It is also robust against false alarms, out-of-order messages and network partitions. The only shortcoming is its reduced failure model. If anyone wants to extend the model should try one of the techniques presented by Neiger and Toueg [29] to automatically translate this algorithm, that only tolerates crash failures, into another one that tolerates more "severe" failures (i.e. arbitrary or byzantine). As future work we are planning to implement the protocol on a network of transputers.

Acknowledgments

L.M.Silva was supported by JNICT under the Programa Ciência (BM-38-90-IA). This work was partially supported by INIC and by JNICT under contract PMCT/C/TIT/451/90.

References

1. F.P.Preparata,G.Metze and R.T.Chien. "On the connection assignment problem of diagnosable systems", IEEE Transactions on Electronic Computing, 16, Dec.1967, pp.848-854
2. S.E.Kreutzer and S.L.Hakimi. "System - Level Fault Diagnosis : A Survey", Euromicro Journal , May 1987, pp.323-330
3. J.G.Kuhl and S.M.Reddy. "Distributed Fault-Tolerance for Large Multi-Processor Systems", Proc. 7th Annual Symposium on Computer Architectures, May 1980,pp. 23-30
4. J.G.Kuhl and S.M.Reddy. "Fault-Diagnosis in Fully Distributed Systems", Proc. FTCS-11, 1981 pp. 100-105
5. S.H.Hosseini, J.G.Kuhl and S.M.Reddy. "On Self-Diagnosis of the Distributed Systems", Proc. FTCS-15, 1985, pp. 30-35
6. S.H.Hosseini, J.G.Kuhl and S.M.Reddy. "A Diagnosis Algorithm for Distributed Computing Systems with Dynamic Failure and Repair", IEEE Trans.on Computers,March 1984,pp.223-233
7. S.E.Kreutzer and S.L.Hakimi. "Distributed Diagnosis and the System User", IEEE Trans. on Computers, January 1988, pp. 71-78
8. N.H. Vaidja and D.K. Pradhan. "System Level Diagnosis : Combining Detection and Location", Proc. FTCS-21, 1991 , pp.488-495
9. A.Baghi and S.L.Hakimi. "An Optimal Algorithm for Distributed System Level Diagnosis", Proc. FTCS-21 , 1991, pp. 214-221
10. R.Bianchini , K.Goodwin and D.S.Nydic. "Pratical Application and Implementation of Distributed System-Level Diagnosis Theory " Proc. FTCS-20 , 1990 , pp. 332-339
11. R.Bianchini and R.Buskens. " An Adaptive Distributed System-Level Diagnosis Algorithm and Its Implementation", Proc. FTCS-21, 1991, pp. 222-229
12. L.Lamport, R.Shostak and M.Pease. "The Byzantine Generals Problem", ACM Trans. on Programming Languages and Systems, V. 4 1982, pp.382-401
13. K.V.S.Ramarao and J.C.Adams. "On the Diagnosis of Byzantine Faults", Proc. of the 7th Symposium on Reliable Distributed Systems , 1988, pp.144-153
14. K.C.Shin and P.Ramanathan. "Diagnosis of Processors with Byzantine Faults in a Distributed Computing System", Proc. FTCS-17, 1987 , pp.55-60
15. J.C.Adams and K.V.S. Ramarao. "Distributed Diagnosis of Byzantine Processors and Links", Proc. 9th Int. Conf. on Distributed Computing Systems, 1989 , pp.562-569
16. R.A. Maxion and F.E. Feather . "A Case Study of Ethernet Anomalies in a Distributed Computing Environment ", IEEE Transactions on Reliability , Vol 39, October 1990 , pp.433-443
17. J.F. Schock et al. " Evolution of the Ethernet Local Computer Network", IEEE Computer , August 1982 , Vol. 5 (8) , pp. 10-27
18. M.L.Powell and D.L.Presoto. "Publishing : A Reliable BroadcastCommunication Mechanism", Proc. 9th Symposium on Operating Systems Principles, 1983 , pp.100-109
19. T. Becker. "Keeping Processes Under Surveillance ", Proc. 10th Symposium on Reliable Distributed Systems,1991,pp.198-205
20. C.I. Dimmer. "The Tandem Non-Stop System", in *Resilient Computing Systems*, Ed. by T. Anderson,1985, pp. 178-196
21. S.Bruso. "A Failure Detection and Notification Protocol for Distributed Computing Systems", Proc. 5th Int. Conf. on Distributed Computing Systems, 1985, pp.116-123
22. B.Walter. "A Robust and Efficient Protocol for Checking the Availability of Remote Sites ", Computer Networks, Vol. 6, 1982 ,pp. 173-188
23. M.Hammer and D.Shipman. "Reliability Mechanisms for SDD-1 : A System for Distributed Databases", ACM Transactions on Database Systems, Vol.5 (4) , Dec. 1980, pp. 431-466
24. F.Cristian. "Agreeing on Who Is Present and Who Is Absent In a Synchronous Distributed System" Proc. FTCS-18, 1988, pp. 206-211
25. H.Kopetz,G.Grunsteidl and J.Reisinger. "Fault-Tolerant Membership Service in a Synchronous Distributed Real-Time System", in *Dependable Computing for Critical Applications*, Springer Verlag, 1991
26. B.Walter. "Network Partitioning and Symmetric Surveillance Protocols", Proc. 5th Int. Conf. on Distributed Computing Systems,85,pp.124 -129
27. A.V.Ma and G.G.Belford."A Failure and Recovery Detection Protocol for Optimistic Partitioned Operation in Distributed Database Systems", Proc. 6th Int. Conf. on Distributed Computing Systems,1986, pp.540-547
28. L.M.Silva , J.G. Silva. "Global Checkpointing for Distributed Programs", Proc. 11th Symp. on Reliable Distributed Systems, pp. 155-162, 1992, Houston (USA)
29. G.Neiger and S.Toueg. " Automatically Increasing the Fault-Tolerance of Distributed Systems", Proc. 7th Symp. on Principles of Distributed Computing, 1988, pp.248-262

APPENDIX 1005D

LIBRARY CATALOG

[Sign in](#) | [Selected Items \(0\)](#) | [Search History](#) | [My Account](#)

Search...

All Fields

Q

Q ADVANCED SEARCH

[Back to catalog results \(287221\)](#)

« Previous | 1 of 2 | [Next](#) »

☐ Select

Cite

Email

Export ▾

Print

Microprocessing and microprogramming

Format:

Journal/Periodical

Language:

English.

Published:

Amsterdam; New York : Published for Euromicro by North-Holland Pub. Co., 1981-

Subject:

[Microprocessors](#) > [Periodicals.](#)
[Microprogramming](#) > [Periodicals.](#)

Description:

v. : ill. ; 26 cm.

ISSN:

0165-6074

Frequency:

Ten numbers per yea

Other contributor:

[Euromicro.](#)

Continues:

Euromicro journal

Notes:

Vol. 7, no. 1 (Jan. 1981)-

Indexes:

v. 1-25, 1975-1989, 1 v.

Availability

Library Annex [Hours](#)

QA76 .5 .E89 [Text](#)

Library has: v.7-41 (1981-1995/1996)

Indexes: v.1/25 (1975/1989)

Notes: Subscription cancelled after 1995.

Available

Request item

Request scan of article/chapter

Other forms of this work

[Microprocessing and microprogramming](#)

Journal/Periodical English Online

[See all forms of this work](#)

Browse related items by call number

QA76.5 .E89

[Expanded view](#)

Q - Science ▶ QA - Mathematics ▶ QA74-76 - Computer Science ▶

QA76.5 .F33

[Online searching : a primer](#)

c1984

Available

QA76.5 .F36

[LUCAS associative array processor : design, programmi...](#)

c1986

Available

QA76.5 .F46

[The logic of computer arithmetic](#)

[1963]

Available

[Microprocessing and microprogramming](#)

Publisher: Published for Euromicro by North-Holland Pub. Co.

Date: 1981-

Available at: Library Annex

Chat with Us

[Librarian View](#)

CONTACT US

 Ask a Librarian

Feedback

RESOURCES

[Library Website](#) | [Search](#) | [Catalog](#) | [Articles & Full Text](#) | [Databases](#) | [E-journal Titles](#) | [Images](#)

[Release Notes](#)

[Credits](#)

Help

Chat with Us

LIBRARY CATALOG

[Sign in](#) | [Selected Items \(0\)](#) | [Search History](#) |  [My Account](#)

All Fields

▼



 **ADVANCED SEARCH**

 [Back to item](#)

Librarian View

LEADER 01075cas a2200325 a 4500	
001 287221	
005 20141105035837.0	
008 810730c19819999ne mn p	0 a0eng d
022	#a 0165-6074
035	#a (CStRLIN)NYCX3047S
035	#a (NIC)notisABF3104
035	#a (OCoLC)7324063
035	#a 287221
040	#a NIC #c NIC
245 0 0	#a Microprocessing and microprogramming.
260	#a Amsterdam; #a New York : #b Published for Euromicro by North-Holland Pub. Co., #c 1981-
300	#a v. : #b ill. ; #c 26 cm.
310	#a Ten numbers per yea
362 0	#a Vol. 7, no. 1 (Jan. 1981)-
555	#a v. 1-25, 1975-1989, 1 v.
650 0	#a Microprocessors #x Periodicals.
650 0	#a Microprogramming #x Periodicals.
650 7	#a Microprocessors #2 fast #0 (OCoLC)fst01020008
650 7	#a Microprogramming #2 fast #0 (OCoLC)fst01020033
655 7	#a Periodicals #2 fast #0 (OCoLC)fst01411641
710 2	#a Euromicro.
780 0 0	#t Euromicro journal
905	#a 19920807120000.0
948 2	#a 20141103 #b m #d batch #e lts #x addfast
950	#l ENGR #a QA76.5 #b .E89+ #i 07/30/81 C ?9777:
999 f f	#i 592b8a9d-92c6-49a9-a00f-2b1b61961d45 #s 3cf10f19-f6a9-4f7e-803c-a5756bac6e6f

CONTACT US

 [Ask a Librarian](#)

[Feedback](#)

RESOURCES

[Library Website](#) | [Search](#) | [Catalog](#) | [Articles & Full Text](#) | [Databases](#) | [E-journal Titles](#) | [Images](#)

[Release Notes](#)

[Credits](#)

APPENDIX 1005E



Library Catalog

McFarlin Library | Mabee Legal Information Center



(Search History)

LEADER 00000nam 22000001 4500
008 930824s1993 xx 000 0 eng d
040 OKT|cOKT
099 Periodicals
111 2 Euromicro Conference on Software Maintenance and
Reengineering|n(19th :|d1993 :|cBarcelona, Spain)
245 10 Proceedings EUROMICRO 93 Open System Design|bHardware,
Software and Applications; Barcelona, September 6-9, 1993.
246 3 EUROMICRO '93
246 3 Open system design
260 Amsterdam, The Netherlands|bNorth-Holland|c1993.
300 xxiii, 861 p.|bill.|c27 cm.
500 In: Microprocessing and microprogramming; v.38, nos.1-5
(single vol.), September 1993.
500 "Nineteenth EUROMICRO conference"
504 Includes bibliographical references.
590 SHELVED AS VOL. 38, NOS. 1-5 OF THE PERIODICAL
"MICROPROCESSING AND MICROPROGRAMMING"
650 0 Computer network protocols|vCongresses.
650 0 Computer network protocols.
650 0 Computer network protocols|xStandards|vCongresses.
710 2 Euromicro.
830 0 Microprocessing and microprogramming ;|vv.38, nos.1-5.
902 010413
907 .b10056804|bmpr|cs
948 LTI 08/31/2001

LOCATION		CALL #		STATUS	
McFarlin Periodicals (Intermediate Level)		Periodicals (See NOTE)		AVAILABLE	

Go to: [The University of Tulsa](#) | [McFarlin Library](#) | [Mabee Legal Information Center](#)

Copyright © 1996-2022 [The University of Tulsa]. All rights reserved.
Today's date: November 15, 2022

McFarlin Library
2933 East 6th Street
Tulsa, Oklahoma 74104-3123
918-631-2871

Mabee Legal Information Center
3120 East Fourth Place
Tulsa, Oklahoma 74104
(918) 631-2404