SOFTWARE REUSE DONOR CARDS

By

LEE S. FIELDS

Bachelor of Science

University of Missouri

Columbia, Missouri

1985

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May 1997

SOFTWARE REUSE DONOR CARDS

Thesis Approved:

Mansur Samadzadeh
Thesis Advisor

Blayne E. Mayfield

W. Dick Stuart

Thomas C. Collins
Dean of the Graduate College

PREFACE

The purpose of this study was to determine the feasibility of designing the equivalent of an organ donor card for software reuse artifacts. Organ donor cards have helped spur the implementation of a national organ transplantation distribution system. Large scale reuse efforts have government or corporate support for the tools and personnel necessary to run a program for reuse-in-the-large, but support for reuse-in-the-small is generally lacking. A simple document, similar to an organ donor card, included with source code (e.g., when distributed over the Internet) would ease reuse for developers not involved with large, well-financed projects.

The design of a software reuse card is proposed using a metadata format. The design includes the equivalent functionality of an organ donor card as well as data specific to software reuse concerns. The proposed reuse donor card was tested using a variety of software artifacts ranging from abstract design patterns to source code fragments and functions. The card is designed to support automation in both generation and retrieval while being simple enough so as to not discourage its use. The donor card was evaluated by a number of professional programmers.

ACKNOWLEDGMENTS

I sincerely thank my advisor Dr. Mansur Samadzadeh for his guidance, help, thoroughness, and patience during this project. The other members of my thesis committee, Drs. Blayne Mayfield, Nick Street, and Mitch Neilsen, were also vital in their support and critique of this work.

I also give my special gratitude to my family and friends for their support during my graduate studies. To Julie for her love, understanding, and giving of her time to allow me to complete this. To my mother for the motivation that only a mother could give. And finally my thanks go to Christy Valliere and Greg Gudenburr for their friendship.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Much of the software reuse effort is expended to enable reuse for large projects with centralized repositories, but this generally ignores a large segment of development projects carried out by small teams or individuals. To encourage software reuse in an unstructured, distributed environment, this thesis presents the design of a software reuse donor card that is analogous to the organ donor card.

Organ donor cards have helped spur the implementation of a national organ transplantation distribution system. This thesis presents an equivalent document for software artifacts to encourage similar developments in software reuse. Large scale reuse efforts have government or corporate support for the tools and personnel necessary to run a program for reuse-in-the-large, but support for reuse-in-the-small is lacking. A simple document, similar to an organ donor card, included with source code or other software assets (e.g., when distributed over the Internet) would ease reuse for developers not involved with large, well-financed, and centralized projects.

The development of organ donation procedures and policies within the United States offers some valuable lessons in this area of research. Using organ donation

procedures as the basis for software reuse seems to be a natural fit. This thesis covers organ donation as it can be related to software reuse and then presents the design of a software reuse donor card.

1.1 Definitions

In this thesis software reuse is defined as using existing software artifacts in the construction of a new software system, through application or incorporation [Dusink and van Katwijk 95]. Software artifacts include any product generated in the development of a software system including but is not limited to: code, classes, object modules, algorithms, design models, processes, documentation, test plans, and applications.

Metadata, which is used to formally define a software reuse donor card, are defined as data describing data and its attributes [FGDC 94].

CHAPTER II

ORGAN DONATION

Upon a person's death, they may bequeath or donate their bodily organs to be made available for transplantation into one or more persons whose organs are failing. This is referred to as organ donation. For certain types of transplants, live donors can give an organ such as a kidney or bone marrow.

2.1 Distribution Organization

Organ donation within the United States is managed at several levels. At the national level, the federal government funds the Organ Procurement and Transplantation Network (OPTN). The OPTN exists to connect the local Organ Procurement Organizations (OPO's) that are typically associated with one or more hospitals [Prottas 94]. It is the responsibility of the local hospitals to identify potential donors, secure consent of the donor or the family of the donor, and to retrieve the organs. This task is handled by the doctors or nurses attending the patient.

2.2 Legal Issues

Consent for organ donation is prioritized in order to manage the liability aspects of organ retrieval [Overcast 87]. The donor's consent has the highest priority and is followed by that of the immediate family members. Organ donation cards are the primary legal instruments used to convey the donor's consent.

The other major legal aspect of organ donation is the limitation of liability. Hospitals and doctors are protected by the Uniform Anatomical Gift Act of 1968, which removes the legal liabilities of organ retrieval [Overcast 87].

The federal government is concerned with the equitability of distribution of organs, so they are considered property of the public. Therefore, it is illegal to buy or sell organs in the United States. However some organs receive special consideration under the law, e.g., corneas are covered under presumed consent laws and may be taken without permission unless expressly forbidden [Prottas 94].

2.3 Donor Benefits

The benefits for the donor and their family include: satisfaction in giving to others, not wasting usable organs, bringing something positive out of death, and having a part of the family member live on [Prottas 94].

2.4 Recipient Benefits

The primary benefit for the recipient is an improved life span or quality of life. Kidney transplants in particular also benefit the federal government by reducing costs when compared with the lifetime cost of dialysis for the patients [Prottas 94].

2.5 Obstacles

The largest problem with organ transplantation is the lack of donors. Organ donor cards are not carried by many people, and when they are, they may not be available at the hospital [Caplan 87]. Even then, doctors do not pursue organ retrieval as often as they should [Prottas 94]. The problem with doctors is not technical in nature, but instead results from the interpersonal difficulties of dealing with families. According to Jeffery Prottas, "willingness is high, communication low".

2.6 Similarities to Software Reuse

The proceeding sections covering organ donation bear a great resemblance to many aspects of software reuse.

- The majority of the organizational structure is local with the federal government providing the national level network to link the localities (Section 2.1). This is similar to the development of the Internet, which started as a federally funded

backbone by first the Department of Defense and then the National Science Foundation to interconnect local network entities [Cerf and Aboba 93].

- The donor and their family derive some consolation from helping others (Section 2.3). The closest software equivalent is advancing the state of the art or altruistically making code available to save others the development effort.
- The recipient benefits from improved quality or span of life (Section 2.4). The corresponding software benefits include improved quality through the use of previously debugged code and decreased development time.
- The biggest obstacles to organ donation are social, not technical issues (Section 2.5). Social issues are also a significant issue for software reuse, such as the "Not Invented Here" syndrome [Contel 91].

CHAPTER III

REUSE

3.1 Reuse Overview

Reuse in software engineering occurs at many levels. It can also be classified along at least three dimensions: type of artifact, scale of artifact, and abstraction level. The NATO reuse standard lists forms of reuse according to the type of artifact at the specification, design, code, test, and documentation levels [Contel 91]. Lajoie and Keller divide object-oriented reuse into three major levels according to scale [Lajoie and Keller 94]: reuse-in-the-large, the reuse of applications, reuse-in-the-medium, the reuse of frameworks or design patterns, and reuse-in-the-small, the reuse of classes or code fragments.

Forms of reuse other than code exist in varying degrees. Algorithms have existed in print for quite some time. However, specifications and documentation available for reuse are scarce.

One form of reuse at a higher level of abstraction than code, which is currently drawing much attention, is design patterns. Design patterns are abstract reuse artifacts

which are "descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context" [Gamma et al. 94]. Design patterns are available in print and on the Internet.

3.2 Distribution Organization

Many companies have successful internal reuse programs. The Contel Corporation and the Department of Defense (DoD) have several programs for large scale national reuse on military projects [Contel 91] [Software 92]. Small scale reuse is currently limited to the use of function libraries that ship with compilers or are available from commercial sources [Prieto-Diaz 93].

A large amount of source code is available from a variety of sources on the Internet or other sources. A few examples are:

The Free Software Foundation's primary GNU site

ftp://prep.ai.mit.edu/pub/gnu

The FreeHEP (Free High Energy Physics) collection site

ftp://freehep.scri.fsu.edu

GAMS: Guide to Available Mathematical Software

http://gams.nist.gov

The main technical problem with reusing such code is the difficulty in locating the proper software component that fits a particular set of requirements [Sembugamoorthy et

al. 92]. Several different retrieval methods have been successful when used with software libraries internal to an organization. Various forms of knowledge-based and full text indexing systems have been studied by researchers [Sembugamoorthy et al. 92], while Prieto-Diaz demonstrated the effectiveness of the faceted classification scheme [Prieto-Diaz 91].

3.3 Nontechnical Issues

Other problems with reuse are nontechnical in nature. There are many legal issues that arise when using someone else's code. For example, the GNU General Public License, which accompanies much of the source code electronically available, requires six pages of text covering the licensing, use, and requirements for software which the user has the freedom to change and redistribute [FSF 91]. Another issue that hinders reuse is the investment required to write reusable code. Code written for reuse incurs additional development costs and requires a substantial amount of discipline on the part of the developer [Prieto-Diaz 93].

3.4 Donor Benefits

Some of the potential benefits to the donors of code or other reusable software artifacts are: satisfaction in giving to others, receiving feedback/improvements to code, licensing fees, and fame. The GNU Project and the Free Software Foundation (FSF)

espouse the idea that sharing code is a direct consequence of the traditional belief "do unto others as you would have them do unto you", and that not sharing code is wrong [Stallman 96].

3.5 Recipient Benefits

The recipient benefits through: increased productivity, reduced maintenance, improved interoperability, improved prototyping capabilities, and reduced training costs [Contel 91]. The code may also have a purely educational benefit. Reading code is rarely practiced, but is an excellent way to learn programming techniques as well as the reasons behind these techniques [Weinberg 71].

3.6 Relation to Organ Donation

When compared to organ donation, software reuse bears many similarities, in both benefits and weaknesses. Some of the similarities were covered earlier (Section 2.6), but the differences are also significant as listed below.

- Unlike organs, code can be legally marketed and reused multiple times.
- Software practitioners do not enjoy the same legal protection as the medical community or the donors.
- Code can be replicated without affecting the original.
- It is preferable that the donor is alive and robust.

- Code reuse is not as extreme (i.e., does not require death or invasive surgery) or as expensive as organ donation.

CHAPTER IV

SOFTWARE REUSE DONOR CARD

4.1 Organ Donor Card

Organ donor cards are a simple method for an individual to convey consent and record choice for the transplantation of their organs upon death.

The sample Uniform Donor Card [Knaus 96] depicted in Figure 1 contains two main categories of information: identification and distribution. The identification information simply identifies who is donating the organs, while the distribution information can be further broken down into what organs to distribute and the legal authorization to distribute.

UNIFORM DONOR CARD

OF _____
(Print or type name of donor)

In the hope that I may help others, I hereby make this anatomical gift, if medically acceptable, to take effect upon my death. The words and marks below indicate my desires:
I give (a)_____any needed organs or tissues
(b)_____only the following organs or tissues

_____
Specify the organ(s) or tissue(s)

for the purpose of transplantation, therapy, medical research or education:
(c)_____my body for anatomical study if needed.

Limitations or special wishes, if any:
_____

Signed by the donor and the following two witnesses in the presence of each other:

_____ _____
Signature of Donor Date of Birth of Donor

_____ _____
Date Signed City and State

_____ _____
Witness Witness
(Preferably next of kin)

This is a legal document under the Uniform Anatomical Gift Act or similar laws.

Figure 1. Uniform Organ Donor Card

Additional information that is not included on the card but necessary for the donation process is tissue typing information that is organic to the donor. In kidney transplantation for example, there are two primary considerations for matching a kidney with a recipient. First, they must have compatible major ABO blood groups. And second, there must be a good matching for antigens of the HLA major histocompatibility gene complex [Carpenter and Lazarus 83].

To better understand the format of existing organ donor cards and to precisely

specify the design of a software reuse donor card, a formal notation can be used. In this situation a textual notation is sufficient since both forms of donor cards are text based. Text operates as the lowest common denominator for these types of documents, and text is easy to manipulate with a variety of tools and on a variety platforms. When looking at existing notations, one that is close in design and intent to the software reuse donor card is the geospatial metadata format developed by the Federal Geographic Data Committee (FGDC) [FGDC 94].

The FGDC is charged with the development of a National Spatial Data Infrastructure (NSDI) [FGDC 97a] in order "to reduce duplication, to reduce the expense of developing geographic data, and to increase the benefits of using available data" [FGDC 97b]. Federal departments and independent entities involved with the FGDC include: the Department of Agriculture, the Department of Commerce, the Department of Defense, the Department of Energy, the Department of Housing and Urban Development, the Department of the Interior, the Department of State, the Department of Transportation, the Environmental Protection Agency, the Federal Emergency Management Agency, the Library of Congress, the National Archives and Records Administration, the National Aeronautics and Space Administration, the Tennessee Valley Authority, the Department of Education, the Department of Health and Human Services, the Department of Justice, the Department of Labor, the General Services Administration, the National Capital Planning Commission, and the Smithsonian Institution [FGDC 97b]. The FGDC member agencies are mandated to develop and follow the FGDC standard for documenting geospatial data [FGDC 97a]. Given the scope of governmental agencies participating in

the standard, it carries significant weight as a standard.

Building upon the overall structure of the FGDC standard, as well as reusing parts of the standard, results in some of the same benefits attributed to software reuse. In this case both risk and development are reduced while taking advantage of the knowledge of experts [Sommerville 92].

One advantage of the FGDC standard is its hierarchical format. Other metadata formats, such as the Dublin Core [Weibel et al. 97] or RFC 1807 [Lasher and Cohen 95] use a flat structure which works well for small sets of metadata but is less readable for larger sets.

The FGDC standard is also machine readable as is demonstrated by the many compilers available to verify and process metadata in this format [Doughty and Van Guilder 97].

The formal description of the organ donor card and the software reuse donor card is given using a metadata format in Appendix B. Mapping this data into a metadata format [FGDC 94] generates the following production rules.

```
Organ_Donor_Card =
   Identification_Information +
   Distribution_Information +
   Implicit_Typing_Information

Identification_Information =
   Name

Distribution_Information =
   Organs +
   Legal_Information

Legal_Information =
   Donor_Signature +
   Donor_Birth_Date +
```

```
    Donor_Signature_Date +
    Donor_Signature_Location +
    Witness_1 +
    Witness_2

Implicit_Typing_Information =
    Blood_Type +
    Tissue_Matching_Information
```

4.2 Software Reuse Donor Card Design

The purpose of an analogous card for software assets is to convey the consent and choice of the donor, and to be an aid for locating suitable code (or any software artifact or software configuration item in general) for reuse.

In contrast to large-scale development, that typically deals with special purpose (i.e., domain specific) code repositories, and institutes and utilizes in-house software reuse, a software reuse donor card would foster reuse across the board and among various development efforts.

At the highest level, a software reuse donor card is similar in structure to an organ donor card. It contains identification information, distribution information, and explicit typing information. Unlike an organ donor card, it also includes reuse information (e.g., licensing information, reuse count, and related artifacts). The top level metadata format for a software reuse donor card is similar to the format of an organ donor card with the addition of an optional section for reuse information.

Software Reuse Donor Card	Organ Donor Card
Software Reuse Card = Identification_Information + Distribution_Information + Typing_Information + (Reuse_Information)	Organ Donor Card = Identification_Information + Distribution_Information + Implicit_Typing_Information

Figure 2. Top Level Metadata Card Comparison

The full software reuse card definition is expanded in Appendix C, but a sample is included here to demonstrate the information available for a typical code module. Appendix D contains a complete template in which to enter the data. An example of a non-code artifact, the Visitor design pattern [Gamma et al. 94], is included as Appendix E to demonstrate the flexibility of the software reuse donor card in documenting a design artifact.

```
Name: STRINGS - String Manipulation Library
Author: Lee S. Fields, Conoco Inc., lee.s.fields@conoco.dupont.com
Description: A library of string manipulation functions for text
   replacement, word parsing, and formatting.
Keywords: string, C, parse, word, column, justify, format, text
Language:
   Name: ANSI C
   Compiler_Information:
      Compiler_Name: IBM C-Set ++
```

```
      Compiler_Version: 2.01
   Compiler_Information:
      Compiler_Name: IBM Visual Age C++
      Compiler_Version: 3.0
   Portability_Index: 5
Platform:
   Operating_System: OS/2
   Operating_System_Version: 3.0
   Processor_Type: Intel 386+
External_Interfaces:
   Used: None
   Provided: None
File/Location: ftp://fieldls.po.dupont.com/distrib/strings.c,
   ftp://fieldls.po.dupont.com/distrib/strings.h
Comments: Written in ANSI C so easily portable to other platforms.
   Written for the TRACERS systems and has been in production use for
   5 years.
Copyright: CONFIDENTIAL AND PROPRIETARY INFORMATION OF CONOCO INC.
   PROTECTED BY THE COPYRIGHT LAW AS AN UNPUBLISHED WORK.
Licensing_information:
   Contact_Information:
      Contact_Type_Primary: Person
      Contact_Person: Lee S. Fields
      Contact_Organization: Conoco Inc.
      Contact_Position: Sr. Systems Analyst
      Contact_Address:
         Address_Type: mailing and physical
         Address: 650-23 ST
         Address: 1000 S. Pine
         City: Ponca City
         State_or_Province: OK
```

```
            Postal_Code: 74601
            Country: USA
         Contact_Voice_Telephone: (405) 767-2192
         Contact_Facsimile_Telephone: (405) 767-3308
         Contact_Electronic_Mail_Address: lee.s.fields@conoco.dupont.com
         Contact_URI: http://hoapsdl.ho.dupont.com/dnstream/dis/tracers
         Hours_of_Service: 8:00 AM - 4:30 PM, Central Time
      License: For Conoco/DuPont internal use only
      License_Restrictions: For Conoco/DuPont internal use only
      License_Cost: N/A
   Date_of_Issue: September 29, 1996
   Version_Information:
      Version: 1.2
      Date_of_This_Version: August 13, 1994
      Release_Frequency: infrequent
      Date_of_Next_Version: N/A
   Reuse:
      Count: 2
      Reused_By: TRACERS ACB Feed system, TRACERS CGI ContractView
   Related_Artifacts:
      Contained_By: TRACERS, ACB Feed, CGI ContractView
      External_Dependencies: PMTOOLS.H memory management functions
```

The reuse card is designed to be written as a plain text file with the file extension .CRD and is to be located with the software artifact that it describes. This locality of reference is similar to the placement of organ donor cards on the back of driver licenses. Giving the card a specific file extension simplifies the process of locating the card with the artifact. This supports searches using tools such as `archie` or Internet search engines. It may also be included in the source code, such as a C source code file or

header file. Another alternative is to implement the card as an HTML file with hypertext links to the referenced artifacts.

Loral Federal Systems has implemented an HTML component search and submission tool for an internal reusable software library [Poulin and Werkman 95]. Their Federal Reuse Repository (FRR) uses seven data items to create a text based Structured Abstract (SA) for components. The SA template follows this format:

> A (Computer Language) (Component Type) for (Domain) that provide (Function) on (Data) data. Runs on (Operating System). Includes (Element, ..., Element). Contact (Contact).

The software reuse donor card contains several of the same data elements, but the application specific elements such as Domain, Function, and Data should be addressed in the description field. The Structured Abstract query tool is a centralized application that maintains the list of all available selections and allows choices only from each list of defined selections.

The HTML 2.0 specification (RFC 1866) includes a META tag "to provide a means to discover that the data set exists and how it might be obtained or accessed; and to document the content, quality, and features of a data set, indicating its fitness for use" [Berners-Lee and Connolly 95]. The use of this tag is not mandatory and its implementation for servers and clients is unspecified. A break-out group at the *W3C Distributed Indexing and Searching Workshop*, Cambridge, MA, May 1996, proposed a convention for embedding structured metadata within HTML documents [Weibel 97]. The proposed convention uses the HTML META tag as follows:

<META NAME = "*schema_identifier.element_name*"

CONTENT = "*string data*">

Adapting this format to software reuse donor cards would use SRDC as the schema_identifier and the element name would include a flattened version of the hierarchical donor card element names. Examples from the "STRINGS" card presented earlier would be formatted as:

```
<META   NAME      = "SRDC.Name"
        CONTENT   = "STRINGS - String Manipulation Library">
<META   NAME      = "SRDC.Language.Name"
        CONTENT   = "ANSI C">
```

Currently, very few tools utilize the HTML META tag. Some but not all search engines will look for META tags with the names of "keywords" or "description", and use their contents to index the document [Richmond and Richmond 97]. Without the development of specialized tools to index and search based on the contents of the META tags, there is little incentive to create tags for any fields other than those that match the software reuse donor card Keywords and Description elements. Encoding the complete donor card both as displayable text and as META tags would be a duplication of effort if done manually. However, a tool to aid in capturing the data for a software reuse donor card could easily store the data as displayable text and as META tags. Appendix G contains a sample software reuse donor card in HTML format using META tags.

The primary goals of the software reuse donor card are:

- Ease of use - if it is not easy to use, it will not be used.
- Not to limit searching capabilities - by including both keywords and a natural language description, differing search philosophies can be supported (the card does

not have to be searchable, but the value of it is greatly reduced if it is not).

- Not to require a central repository or support staff - this allows the full spectrum of developers to benefit.
- To provide enough information to be self-contained - from the card a potential user should know hardware, software, and legal requirements for the use of the artifact.
- To be structured to support automation - allows automated creation or searching.

While some of these goals can be considered conflicting (such as ease of use versus being self-contained) the current design does meet these goals. The flexibility available in the metadata format permits the creator of a reuse card to include just the minimum data required or add the extra data that further benefits the artifact's potentials reusers. Appendix F outlines the proposed minimal software reuse donor card.

CHAPTER V

EVALUATION

5.1 Peer Review

To evaluate the design of the software reuse donor card, the opinions of four professional software developers were solicited. Two of the developers had 16 years of experience while the other two had 6 and 15 years of experience in professional software development. All of the four developers had experience in both mainframe as well as client-server applications.

The developers reviewed the design individually, but their evaluations were similar. The consensus was that software reuse donor cards would be useful, but only if they were available electronically. The reviews stated that software reuse donor cards needed to be on the web (World Wide Web) or a central repository to enable searching.

Their opinions on the importance of creating software reuse donor cards varied with their perceived motivation for doing so. One developer stated that they would create the donor cards, "but probably minimal information would be entered." Another

developer stated, "If it would help market my software I would do it." The third developer proposed that they might start with donor cards early in development to make sure the resulting code was reusable, but that developers tend "to code, implement, and wash their hands of the system" and rarely follow up development with "documentation" like this. The fourth developer commented that they would only complete the form if it was in an electronic format which must be queryable.

The only comment received specific to one of the donor card elements concerned the portability index. One developer felt that generating the value for the portability index should be automated otherwise it would be subjective to the donor card creator's experience or lack of experience in porting code.

5.2 Comparison to Other Repositories

To compare the design of software reuse donor cards with existing reuse repositories, I selected a sample group of four systems designed for access through the Internet. These systems and representative entries are:

Asset Source for Software Engineering Technology (ASSET) [ASSET 97],

http://www.asset.com/WSRD/abstracts/ABSTRACT_1324.html

Public Ada Library (PAL) Card Catalog [PAL 97],

http://wuarchive.wustl.edu/languages/ada/userdocs/html/cardcat/toolkit.html

CMU Artificial Intelligence Repository, and

http://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/lang/lisp/gui/clx/0.html

Structured Abstracts

[Poulin and Werkman 95]

The level of information provided by these systems is equivalent or slightly greater than that defined as the minimum software reuse donor card in Appendix F. All of the existing comparison systems rely upon a centralized repository, and three out of four provide only limited searching by keyword or an index. One system, PAL, includes extensive information on the file listings for each entry that far exceeds that contained within the software reuse donor card design.

CHAPTER VI

SUMMARY AND FUTURE WORK

The objective of this thesis was to ascertain the feasibility of designing a software reuse donor card to support reuse in a distributed, unstructured development environment. It needed to meet conflicting goals of functionality versus ease of use while supporting a broad range of software artifacts. Validation by peer review and by comparison to other existing reuse repositories supports the assertion that the proposed design successfully meets these criteria.

Future work may include: design validation with expanded testing, providing a facility to automate the creation of reuse cards (either after-the-fact or in tandem with the software development environment), research the organization and searching of reuse cards in a distributed environment, and conducting a usability study to measure the acceptance of the proposed design.

REFERENCES

[Berners-Lee 94] T. Berners-Lee, "Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as Used in the World-Wide Web", RFC 1630, CERN, June 1994.

[Berners-Lee and Connolly 95] T. Berners-Lee and D. Connolly, "Hypertext Markup Language - 2.0", ftp://ds.internic.net/rfc/rfc1866.txt (28 January 1997).

[Caplan 87] Arthur L. Caplan, "Obtaining and Allocating Organs for Transplantation", *Human Organ Transplantation*, Edited by: Dale H. Cowan, Jo Ann Kantorowitz, Jay Moskowitz, and Peter H. Rheinstein, Health Administration Press, Ann Arbor, MI, 1987.

[Carpenter and Lazarus 83] Charles B. Carpenter and J. Michael Lazarus, "Dialysis and Transplantation in the Treatment of Renal Failure", *Harrison's Principles of Internal Medicine*, Edited by: Robert G. Petersdorf, Raymond D. Adams, Eugene Braunwald, Kurt J. Isselbacher, Joseph B. Martin, and Jean D. Wilson, McGraw-Hill Book Company, New York, NY, 1983.

[Cerf and Aboba 93] Vinton Cerf and Bernard Aboba, "How the Internet Came to Be", gopher://gopher.isoc.org:70/00/internet/history/how.internet.came.to.be (2 October 1996).

[Contel 91] Contel Corporation, *NATO Standard for the Development of Reusable Software Components, Volume 1*, NATO contract number CO-5957-ADA, 1991.

[Doughty and Van Guilder 97] Jonathon Doughty and James Van Guilder, "Metadata Tool Evaluation", http://www.fgdc.gov:80/Metadata/Mitre/task2/index.html (22 January 1997).

[Dusink and van Katwijk 95] Liesbeth Dusink and Jan van Katwijk, "Reuse Dimensions", *Proceedings of the ACM SIGSOFT Symposium on Software Reusability*, Edited by: Mansur H. Samadzadeh and Mansour K. Zand, Seattle, WA, pp. 137-149, April 1995.

[FGDC 94] Federal Geographic Data Committee, "Content Standards for Digital

Geospatial Metadata" (June 8, 1994), Federal Geographic Data Committee, Washington, D.C., 1994.

[FGDC 97a] Federal Geographic Data Committee, "Executive Order 12906", http://www.fgdc.gov/execord.html (15 January 1997).

[FGDC 97b] Federal Geographic Data Committee, "FGDC Manual of Federal Geographic Data Products - Overview", http://www.usgs.gov/fgdc-catalog/overview.html (15 January 1997).

[FSF 91] Free Software Foundation, *GNU General Public License, Version 2*, 1991.

[Gamma et al. 94] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley Publishing Company, Reading, MA, 1994.

[Knaus 96] Dave Knaus, "Uniform Organ Donor Card", http://www.execpc.com/~wa9pov/donorwp5.zip (1 October 1996).

[Lajoie and Keller 94] Richard Lajoie and Rudolf K. Keller, "Design and Reuse in Object-Oriented Frameworks: Patterns, Contracts, and Motifs in Concert", *Proceedings of the 62nd Congress of the Association Canadienne Française pour l'Avancemet des Sciences* (ACFAS), Montreal, Canada, May 1994.

[Lasher and Cohen 95] R. Lasher and D. Cohen, "A Format for Bibliographic Records", http://ds.internic.net/rfc/rfc1807.txt (22 January 1997).

[Overcast 87] Thomas D. Overcast, "Legal Aspects of Death and Informed Consent in Organ Transplantation", *Human Organ Transplantation*, Edited by: Dale H. Cowan, Jo Ann Kantorowitz, Jay Moskowitz, and Peter H. Rheinstein, Health Administration Press, Ann Arbor, MI, 1987.

[Poulin and Werkman 95] Jeffrey S. Poulin and Keith J. Werkman, "Melding Structured Abstracts and the World Wide Web for Retrieval of Reusable Components", *Proceedings of the ACM SIGSOFT Symposium on Software Reusability*, Edited by: Mansur H. Samadzadeh and Mansour K. Zand, Seattle, WA, pp. 137-149, April, 1995.

[Prieto-Diaz 91] Ruben Prieto-Diaz, "Implementing Faceted Classification for Software Reuse", *Communications of the ACM*, vol. 34, no. 5, pp. 88-97, May 1991.

[Prieto-Diaz 93] Ruben Prieto-Diaz, "Status Report: Software Reusability", *IEEE Software*, vol. 10, no. 3, pp. 61-66, May 1993.

[Prottas 94] Jeffery Prottas, *The Most Useful Gift*, Jossey-Bass Publishers, San Francisco,

CA, 1994.

[Richmond and Richmond 97] Alan Richmond and Lucy Richmond, "META Tagging for Search Engines", http://www.stars.com/Search/Meta.html (21 January 1997).

[Sembugamoorthy et al. 92] Vel Sembugamoorthy, Lynn Streeter, Bill Keese, and Mary Leland, "Igrep: A Real World Perspective on Locating Software Artifacts for Reuse", *Proceedings of the Fifth Annual Workshop on Institutionalizing Software Reuse (WISR5, 1992)*, http://rbse.jsc.nasa.gov/eichmann/wisr/wisr5.html (17 Februrary 1997).

[Software 92] Software Productivity Consortium Services Corporation, *Reuse Adoption Guidebook, Version 01.00.03*, Software Productivity Consortium Services Corporation, Herndon, VA, 1992.

[Sommerville 92] Ian Sommerville, *Software Engineering*, Addison-Wesley Publishing Company, Reading, MA, 1992.

[Stallman 96] Richard Stallman, "The GNU Manifesto", http://www.gnu.org/gnu/manifesto.html (15 October 1996).

[Weibel et al. 97] Stuart Weibel, Jean Godby, Eric Miller, Ron Daniel, "OCLC/NCSA Metadata Workshop Report", http://www.oclc.org:5046/oclc/research/conferences/metadata/dublin_core_report.html (20 January 1997).

[Weibel 97] Stuart Weibel, "A Proposed Convention for Embedding Metadata in HTML", http://www.w3.org/pub/WWW/Search/9605-Indexing-Workshop/ReportOutcomes/S6Group2.html (21 January 1997).

[Weinberg 71] Gerald M. Weinberg, *The Psychology of Computer Programming*, Van Nostrand Reinhold Company, New York, NY, 1971.

APPENDIX A: GLOSSARY

ABO - The three common alleles (A, B, and O) used for human blood grouping.

DoD - Department of Defense

FGDC - Federal Geographic Data Committee

FSF - Free Software Foundation

FRR - Federal Reuse Repository

GNU - Gnu's Not Unix

HLA - Human histocompatibilty surface antigens involved with organ acceptance/rejection

HTML - HyperText Markup Language

NATO - North Atlantic Treaty Organization

NSDI - National Spatial Data Infrastructure

OPO - Organ Procurement Organization

OPTN - Organ Procurement and Transplantation Network

Organ Donor Card - A legally binding document that conveys the individual's desire to donate useful organs upon death for transplantation.

RFC - Request For Comments

SA - Structured Abstract

SRDC - Software Reuse Donor Card

URI - Universal Resource Identifier

W3C - World Wide Web Consortium

WWW - World Wide Web

APPENDIX B: METADATA ORGANIZATION

The style of metadata used in this thesis follows from the Federal Geographic Data Committee's standard for geospatial metadata [FGDC 94]. This standard defines the organizational hierarchy of data elements and compound elements along with the production rules that govern their composition. The productions rules specify how the data elements and compound elements are combined to create higher level compound elements.

Four attributes are used to describe a data element. The data elements name and definition are given, followed by the data type of the element and the domain of valid values the data element may contain.

Four attributes also define a compound element. Like the data element, the name and definition are given first. The data type is always specified as "compound", followed by the data elements for the compound elements that it contains.

From the "Content Standards for Digital Geospatial Metadata" [FGDC 94] the following symbol mappings and element definitions formats are used:

```
Data element name -- definition
   Type:
   Domain:
```

Figure B-1. Data Element Format Specification

```
Compound element name -- definition
   Type:  compound
```

Figure B-2. Compound Element Format Specification

Symbol	Meaning
=	is replaced by, produces, consists of
+	and
[|]	selection - select one term from the list of enclosed terms (exclusive or). Terms are separated by "|"
m{}n	iteration - the term(s) enclosed is(are) repeated from "m" to "n" times
()	optional - the term(s) enclosed is(are) optional

Figure B-3. Symbols in Metadata Production Rules

APPENDIX C: SOFTWARE REUSE DONOR CARD METADATA DEFINITION

The formatting standard used by the Federal Geographic Data Committee [FGDC 94] is used throughout this document.

```
Software Reuse Card =
    Identification_Information +
    Typing_Information +
    Distribution_Information +
    (Reuse_Information)

Identification_Information =
    Name +
    Author +
    Description +
    Keywords +
    Version_Information

Version_Information =
    Version +
    0{Date_of_This_Version}1 +
    0{Release_Frequency}1 +
    0{Date_of_Next_Version}1

Typing_Information =
    Language_Information +
    0{Platform_Information}n +
    (External_Interfaces_Information)

Distribution_Information =
    File_Reference_URI +
```

```
    (Copyright) +
    Licensing_Information

Reuse_Information =
    (Comments) +
    0{Date_of_Issue}1 +
    0{Reuse_Count}1 +
    0{Reused_By}1 +
    0{Related_Artifacts}1 +
    0{Other}1

Language_Information =
    Language_Name +
    (0{Compiler_Information}n) +
    (Portability_Index)

Compiler_Information =
    Compiler_Name +
    (0{Compiler_Version}n)

Platform_Information =
    Operating_System +
    (0{Operating_System_Version}n) +
    (0{Processor_Type}n)

External_Interfaces_Information =
    (0{External_Interfaces_Used}n) +
    (0{External_Interfaces_Provided}n)

Licensing_Information =
    (Contact_Information) +
    License +
    (License_Restrictions) +
    (License_Cost)

Related_Artifacts =
    0{Contains}n +
    0{Contained_By}n +
    0{External_Dependencies}n +
    0{External_Dependents}n +
    0{Derived_From}n +
    0{Derived_By}n +
    0{Adapted_From}n +
    0{Adapted_By}n +
```

```
    0{Inherits_From}n +
    0{Inherited_By}n +
    0{Other_Relations}n

Contact_Information =
    Contact_Type_Primary +
    Contact_Person +
    Contact_Organization +
    (Contact_Position) +
    1{Contact_Address}n +
    (1{Contact_Voice_Telephone}n) +
    (1{Contact_TDD/TTY_Telephone}n) +
    (1{Contact_Facsimile_Telephone}n) +
    (1{Contact_Electronic_Mail_Address}n) +
    (1{Contact_URI}n) +
    (Hours_of_Service) +
    (Contact_Instructions)

Contact_Address =
    Address_Type +
    0{Address}n +
    City +
    State_or_Province +
    Postal_Code +
    (Country)
```

Each data element in this definition consists of its name, description, type, and domain. An element may be a compound type consisting of an aggregation of other data elements.

I. Identification_Information - information to identify a software artifact uniquely.

A. Type: compound.

B. Name - an identifying name for a software artifact.

1. Type: text

2. Domain: free text

C. Author - the author's name and contact information (mail address and/or email address and/or phone number).

1. Type: text

2. Domain: free text

D. Description - a natural language description of the purpose or functionality of the software artifact (keep relevant to artifact to enhance searching).

1. Type: text

2. Domain: free text

E. Keywords - a small set of descriptive keywords to aid in searching.

1. Type: text

2. Domain: free text

F. Version_Information- artifact version identifier.

1. Type: compound

2. Version - artifact version identifier.

a. Type: text

b. Domain: free text

3. Date_of_This_Version - release date for this version of the artifact.

a. Type: text

b. Domain: date

4. Release_Frequency - expected or previous frequency of artifact releases.

    a. Type: text

    b. Domain: free text

5. Date_of_Next_Version - expected release date for the next version of the artifact.

    a. Type: text

    b. Domain: date

II. Typing_Information - information for matching the selected software artifact with requirements.

A. Type: compound

B. Language_Information - the programming language or tool with which the artifact was created.

1. Type: compound

2. Language Name - the name of the programming language or tool with which the artifact was created.

    a. Type: text

    b. Domain: free text

3. Compiler Information - information identifying the target compiler.

    a. Type: compound

    b. Compiler Name - the name of the specific language compiler or tool with which the artifact was created.

        (1) Type: text

(2) Domain: free text

c. Compiler Version - the version identifier for the compiler or creation tool.

(1) Type: text

(2) Domain: free text

4. Portability Index - a measure of the estimated portability of the software artifact. A range of five values was chosen to provide a rough estimate without being overly complicated and unreliable. The values are their associated interpretations are:

1 - Not portable

2 - Major changes required to port

3 - Moderate changes required to port

4 - Minor changes required to port

5 - No changes required to port

a. Type: Integer

b. Domain: 1 (least portable) <= Portability Index <= 5 (most portable)

C. Platform Information - the hardware/software platform for which the artifact is targeted.

1. Type: compound

2. Operating System - the name of the targeted operating system.

a. Type: text

b. Domain: free text

3. Operating System Version - the version identifier for the targeted operating

system.

a. Type: text

b. Domain: free text

4. Processor Type - the name of the targeted processor class.

a. Type: text

b. Domain: free text

D. External Interface Information - external programming interfaces used or provided by the artifact.

1. Type: compound

2. External Interfaces Used - the name of any programming interface external to the software artifact that is used by the software artifact.

a. Type: text

b. Domain: free text

3. External Interfaces Provided - the name of any programming interface provided by the software artifact to other components.

a. Type: text

b. Domain: free text

III. Distribution Information

A. Type: compound

B. File Reference URI - the file(s) name(s) and location specified in URI (Universal Resource Identifier) form [Berners-Lee 94].

1. Type: text

2. Domain: URI

C. Copyright - copyright information about the artifact.

1. Type: text

2. Domain: free text

D. Licensing Information - information or reference to information about licensing the artifact which can include: usage restrictions, cost, contact information, expiration date, etc.

1. Type: compound

2. Contact Information - Adopted from [FGDC 94].

a. Type: compound

b. Contact Type Primary - Is the primary contact a person or an organization.

(1) Type: text

(2) Domain: "Person", "Organization"

c. Contact Person - the name of the contact person.

(1) Type: text

(2) Domain: free text

d. Contact Organization - the name of the contact organization.

(1) Type: text

(2) Domain: free text

e. Contact Position - the position of the contact person.

(1) Type: text

(2) Domain: free text

f. Contact Address - the address of the primary contact.

(1) Type: compound

(2) Address Type - is the address a mailing or physical address.

(a) Type: text

(b) Domain: "Mailing address", "Physical address", "Mailing and physical address"

(3) Address - address text.

(a) Type: text

(b) Domain: free text

(4) City - the city name for the contact address.

(a) Type: text

(b) Domain: free text

(5) State or Province - the state or province of the contact address.

(a) Type: text

(b) Domain: free text

(6) Postal Code - the ZIP or other code for the contact address.

(a) Type: text

(b) Domain: free text

(7) Country - the country of the contact address.

(a) Type: text

(b) Domain: free text

g. Contact Voice Telephone - the voice phone number for the contact person

or organization.

(1) Type: text

(2) Domain: free text

h. Contact TDD/TTY Telephone - the phone number for hearing impaired access to the contact person or organization.

(1) Type: text

(2) Domain: free text

i. Contact Facsimile Telephone - the fax phone number for the contact.

(1) Type: text

(2) Domain: free text

j. Contact Electronic Mail Address - the address for sending electronic mail to the contact.

(1) Type: text

(2) Domain: free text

k. Contact URI - the Universal Resource Identifier for the contact.

(1) Type: text

(2) Domain: free text

l. Hours of Service - the time frame for which the contact is available.

(1) Type: text

(2) Domain: free text

m. Contact Instructions - special instructions for communicating with the contact.

(1) Type: text

(2) Domain: free text

3. License - the license applied to the artifact.

a. Type: text

b. Domain: free text

4. License Restrictions - conditions governing use of the artifact.

a. Type: text

b. Domain: free text

5. License Cost - costs associated with licensing the artifact.

a. Type: text

b. Domain: free text

IV. Reuse Information

A. Type: compound

B. Comments - additional comments about the artifact.

1. Type: text

2. Domain: free text

C. Date of Issue - date the software reuse card was issued.

1. Type: date

2. Domain: free date

D. Reuse Count - the number of additional times the artifact has been reused.

1. Type: text

2. Domain: free text

E. Reused By - other artifacts which have reused the current artifact.

1. Type: text

2. Domain: free text

F. Related Artifacts - other software artifacts to which the current artifact is related.

1.Type: compound

2. Contains - other artifacts contained by source within the current artifact (source code containment)

a. Type: text

b. Domain: free text

3. Contained By - other artifacts which contain the current artifact by source

a. Type: text

b. Domain: free text

4. External Dependencies - other external artifacts required to use the current artifact (e.g. object modules, libraries)

a. Type: text

b. Domain: free text

5. External Dependents - other external artifacts which require the current artifact

a. Type: text

b. Domain: free text

6. Derived From - external artifacts at a higher level of abstraction from which the current artifact was developed (e.g. design models, patterns)

a. Type: text

b. Domain: free text

7. Derived By - external artifacts at a lower level of abstraction which were developed from the current artifact

   a. Type: text

   b. Domain: free text

8. Adapted From - external artifacts from which the current artifact was developed by modification or adaptation

   a. Type: text

   b. Domain: free text

9. Adapted By - external artifacts which were developed by modification or adaptation from the current artifact

   a. Type: text

   b. Domain: free text

10. Inherits From - external artifacts from which the current artifact is created by inheritance

    a. Type: text

    b. Domain: free text

11. Inherited By - external artifacts which are subclasses of the current artifact

    a. Type: text

    b. Domain: free text

11. Other Relations - related artifacts not covered by the preceding relationships

    a. Type: text

b. Domain: free text

G. Other - other information not covered by the preceding categories.

1. Type: text

2. Domain: free text

APPENDIX D: SOFTWARE REUSE DONOR CARD TEMPLATE

Name:

Author:

Description:

Keywords:

Language:

    Name:

    Compiler_Information:

        Compiler_Name:

        Compiler_Version:

    Portability_Index:

Platform:

    Operating_System:

    Operating_System_Version:

    Processor_Type:

External_Interfaces:

    Used:

    Provided:

File/Location:

Comments:

Copyright:

Licensing_Information:

Contact_Information:

Contact_Type_Primary:

Contact_Person:

Contact_Organization:

Contact_Position:

Contact_Address:

Address_Type:

Address:

City:

State_or_Province:

Postal_Code:

Country:

Contact_Voice_Telephone:

Contact_TDD/TTY_Telephone:

Contact_Facsimile_Telephone:

Contact_Electronic_Mail_Address:

Contact_URI:

Hours_of_Service:

Contact_Instructions:

License:

License_Restrictions:

License_Cost:

Date_of_Issue:

Version_Information:

Version:

Date_of_This_Version:

Release_Frequency:

Date_of_Next_Version:

Reuse:

Count:

Reused_By:

Related_Artifacts:

Contains:

Contained_By:

External_Dependencies:

External_Dependents::

Derived_From:

Derived_By:

Adapted_From:

Adapted_By:

Inherits_From:

Inherited_By:

Other_Relations:

Other:

APPENDIX E: SAMPLE ABSTRACT ASSET SOFTWARE REUSE DONOR CARD

Name: Visitor Pattern

Author: Eric Gamma, et al.

Description: A design pattern that models behavior to be applied to an object structure containing multiple classes of objects.

Keywords: design pattern, visitor, multiple classes, composite, double dispatch

Language:

Name: pattern language

File/Location: Eric Gamma, et al., "Design Patterns: Elements of Reusable Object-Oriented Software", Addison-Wesley Publishing Company, Reading, MA, 1994

Comments: The book includes an example in C++ and also references a Smalltalk-80 version.

Copyright: none

Licensing_Information:

License: no license required for the use of the design pattern, however the description of the pattern within the book follows standard copyright law.

Date_of_Issue: October 3, 1996

Version_Information:

Version: unknown

Date_of_This_Version: 1995

Reuse:

Count: Unknown (many)

Reused_By: Smalltalk-80, IRIS Inventor, Fresco Application Toolkit, etc.

Related_Artifacts:

Other_Relations: Composite Pattern, Interpreter Pattern

APPENDIX F: MINIMUM SOFTWARE REUSE DONOR CARD

Taking the definition specified in Appendix C and eliminating all the optional data elements yields this minimal software reuse donor card. It contains only the necessary elements to identify, locate, and describe a software artifact.

Name:

Author:

Description:

Keywords:

Language:

    Name:

File/Location:

Licensing_Information:

    License:

Version_Information:

    Version:

APPENDIX G: SAMPLE HTML FORMAT FOR THE PROPOSED SOFTWARE REUSE DONOR CARD

```
<html>
<head><title>Software Reuse Donor Card - STRINGS</title>
<!-- First include keywords and description META tags for indexing -->
<META   NAME     = "keywords"
        CONTENT    = "string, C, parse, word, column, justify, format, text,
                      SRDC">
<META   NAME     = "description"
        CONTENT    = "A library of string manipulation functions for text
                      replacement, word parsing, and formatting.">
<!-- Now include donor card in META tags -->
<META   NAME     = "SRDC.Name"
        CONTENT    = "STRINGS - String Manipulation Library">
<META   NAME     = "SRDC.Author"
        CONTENT    = "Lee S. Fields, Conoco Inc.,
                      lee.s.fields@conoco.dupont.com">
<META   NAME     = "SRDC.Description"
        CONTENT    = "A library of string manipulation functions for text
                      replacement, word parsing, and formatting.">
<META   NAME     = "SRDC.Keywords"
        CONTENT    = "string, C, parse, word, column, justify, format, text">
<META   NAME     = "SRDC.Language.Name"
        CONTENT    = "ANSI C">
<META   NAME     = "SRDC.Language.Compiler_Information.Compiler_Name"
        CONTENT    = "IBM C-Set++">
<META   NAME     = "SRDC.Language.Compiler_Information.Compiler_Version"
        CONTENT    = "2.01">
<META   NAME     = "SRDC.Language.Compiler_Information.Compiler_Name"
        CONTENT    = "IBM Visual Age C++">
<META   NAME     = "SRDC.Language.Compiler_Information.Compiler_Version"
        CONTENT    = "3.0">
<META   NAME     = "SRDC.Language.Portability_Index"
        CONTENT    = "5">
```

```
<META   NAME    = "SRDC.Platform.Operating_System"
        CONTENT   = "OS/2">
<META   NAME    = "SRDC.Platform.Operating_System_Version"
        CONTENT   = "3.0">
<META   NAME    = "SRDC.Platform.Processor_Type"
        CONTENT   = "Intel 386+">
<META   NAME    = "SRDC.External_Interfaces.Used"
        CONTENT   = "None">
<META   NAME    = "SRDC.External_Interfaces.Provided"
        CONTENT   = "None">
<META   NAME    = "SRDC.File/Location"
        CONTENT   = "ftp://fieldls.po.dupont.com/distrib/strings.c,
                     ftp://fieldls.po.dupont.com/distrib/strings.h">
<META   NAME    = "SRDC.Comments"
        CONTENT   = "Written in ANSI C so easily portable to other platforms.
                     Written for the TRACERS systems and has been in
                     production use for 5 years.">
<META   NAME    = "SRDC.Copyright"
        CONTENT   = "CONFIDENTIAL AND PROPRIETARY
                     INFORMATION OF CONOCO INC. PROTECTED BY
                     THE COPYRIGHT LAW AS AN UNPUBLISHED
                     WORK.">
<META   NAME    =
        "SRDC.Licensing_Information.Contact_Information.Contact_Type_Primary"
        CONTENT   = "Person">
<META   NAME    =
        "SRDC.Licensing_Information.Contact_Information.Contact_Person"
        CONTENT   = "Lee S. Fields">
<META   NAME    =
        "SRDC.Licensing_Information.Contact_Information.Contact_Organization"
        CONTENT   = "Conoco Inc.">
<META   NAME    =
        "SRDC.Licensing_Information.Contact_Information.Contact_Position"
        CONTENT   = "Sr. Systems Analyst">
<META   NAME    =
"SRDC.Licensing_Information.Contact_Information.Contact_Address.Address_Type"
        CONTENT   = "mailing and physical">
<META   NAME    =
      "SRDC.Licensing_Information.Contact_Information.Contact_Address.Address"
        CONTENT   = "650-23ST">
<META   NAME    =
      "SRDC.Licensing_Information.Contact_Information.Contact_Address.Address"
        CONTENT   = "1000 S. Pine">
<META   NAME    =
```

```
      "SRDC.Licensing_Information.Contact_Information.Contact_Address.City"
          CONTENT    = "Ponca City">
<META   NAME    =
"SRDC.Licensing_Information.Contact_Information.Contact_Address.State_or_Province
"
          CONTENT    = "OK">
<META   NAME    =
   "SRDC.Licensing_Information.Contact_Information.Contact_Address.Postal_Code"
          CONTENT    = "74601">
<META   NAME    =
      "SRDC.Licensing_Information.Contact_Information.Contact_Address.Country"
          CONTENT    = "USA">
<META   NAME    =
      "SRDC.Licensing_Information.Contact_Information.Contact_Voice_Telephone"
          CONTENT    = "(405) 767-2192">
<META   NAME    =
   "SRDC.Licensing_Information.Contact_Information.Contact_Facsimile_Telephone"
          CONTENT    = "(405) 767-3308">
<META   NAME    =
"SRDC.Licensing_Information.Contact_Information.Contact_Electronic_Mail_Address"
          CONTENT    = "lee.s.fields@conoco.dupont.com">
<META   NAME    =
      "SRDC.Licensing_Information.Contact_Information.Contact_URI"
          CONTENT    = "http://hoapsd1.ho.dupont.com/dnstream/dis/tracers">
<META   NAME    =
             "SRDC.Licensing_Information.Contact_Information.Hours_of_Service"
          CONTENT    = "8:00 AM - 4:30 PM, Central Time">
<META   NAME    = "SRDC.Licensing_Information.License"
          CONTENT    = "For Conoco/DuPont internal use only">
<META   NAME    = "SRDC.Licensing_Information.License_Restrictions"
          CONTENT    = "For Conoco/DuPont internal use only">
<META   NAME    = "SRDC.Licensing_Information.License_Cost"
          CONTENT    = "N/A">
<META   NAME    = "SRDC.Date_of_Issue"
          CONTENT    = "September 29, 1996">
<META   NAME    = "SRDC.Version_Information.Version"
          CONTENT    = "1.2">
<META   NAME    = "SRDC.Version_Information.Date_of_This_Version"
          CONTENT    = "August 13, 1994">
<META   NAME    = "SRDC.Version_Information.Release_Frequency"
          CONTENT    = "infrequent">
<META   NAME    = "SRDC.Version_Information.Date_of_Next_Version"
          CONTENT    = "N/A">
<META   NAME    = "SRDC.Reuse.Count"
```

```
          CONTENT    = "2">
<META     NAME       = "SRDC.Reuse.Reused_By"
   CONTENT    = "TRACERS ACB Feed system, TRACERS CGI ContractView">
<META     NAME       = "SRDC.Related_Artifacts.Contained_By"
          CONTENT    = "TRACERS, ACB Feed, CGI ContractView">
<META     NAME       = "SRDC.Related_Artifacts.External_Dependencies"
          CONTENT    = "PMTOOLS.H memory management functions">
</head>
<body>
<pre>
Name: STRINGS - String Manipulation Library
Author: Lee S. Fields, Conoco Inc.
Description: A library of string manipulation functions for text replacement, word
   parsing, and formatting.
Keywords: string, C, parse, word, column, justify, format, text
Language:
   Name: ANSI C
   Compiler_Information:
      Compiler_Name: IBM C-Set ++
      Compiler_Version: 2.01
   Compiler_Information:
      Compiler_Name: IBM Visual Age C++
      Compiler_Version: 3.0
   Portability_Index: 5
Platform:
   Operating_System: OS/2
   Operating_System_Version: 3.0
   Processor_Type: Intel 386+
External_Interfaces:
   Used: None
   Provided: None
File/Location: <a href="ftp://fieldls.po.dupont.com/distrib/strings.c">
   ftp://fieldls.po.dupont.com/distrib/strings.c</a>, <a
   href="ftp://fieldls.po.dupont.com/distrib/strings.h">
   ftp://fieldls.po.dupont.com/distrib/strings.h</a>
Comments: Written in ANSI C so easily portable to other platforms.  Written for the
   TRACERS systems and has been in production use for 5 years.
Copyright: CONFIDENTIAL AND PROPRIETARY INFORMATION OF CONOCO
   INC. PROTECTED BY THE COPYRIGHT LAW AS AN UNPUBLISHED
   WORK.
Licensing_Information:
   Contact_Information:
      Contact_Type_Primary: Person
      Contact_Person: Lee S. Fields
```

```
        Contact_Organization: <a href="http://www.conoco.com">Conoco Inc.</a>
        Contact_Position: Sr. Systems Analyst
        Contact_Address:
            Address_Type: mailing and physical
            Address: 650-23 ST
            Address: 1000 S. Pine
            City: Ponca City
            State_or_Province: OK
            Postal_Code: 74601
            Country: USA
        Contact_Voice_Telephone: (405) 767-2192
        Contact_Facsimile_Telephone: (405) 767-3308
        Contact_Electronic_Mail_Address: <a
            href="mailto:lee.s.fields@conoco.dupont.com">
            lee.s.fields@conoco.dupont.com </a>
        Contact_URI: <a href="http://hoapsd1.ho.dupont.com/dnstream/dis/tracers">
            http://hoapsd1.ho.dupont.com/dnstream/dis/tracers</a>
        Hours_of_Service: 8:00 AM - 4:30 PM, Central Time
    License: For Conoco/DuPont internal use only
    License_Restrictions: For Conoco/DuPont internal use only
    License_Cost: N/A
Date_of_Issue: September 29, 1996
Version_Information:
    Version: 1.2
    Date_of_this_version: August 13, 1994
    Release_Frequency: infrequent
    Date_of_next_ version: N/A
Reuse:
    Count: 2
    Reused_By: TRACERS ACB Feed system, TRACERS CGI ContractView
Related_Artifacts:
    Contained_By: TRACERS, ACB Feed, CGI ContractView
    External_Dependencies: PMTOOLS.H memory management functions
</pre>
</body>
</html>
```

VITA

Lee S. Fields

Candidate for the Degree of

Master of Science

Thesis: SOFTWARE REUSE DONOR CARDS

Major Field: Computer Science

Biographical:

Personal Data: Born in Lebanon, Missouri, on April 21, 1961, the son of Robert and Sallie Fields.

Education: Graduated from Jefferson City High School, Jefferson City, Missouri, in May 1979; attended the Massachusetts Institute of Technology from September 1979 until December 1980 studying Mechanical Engineering; received Bachelor of Science degree in Computer Science from the University of Missouri, Columbia, Missouri in May 1985. Completed the requirements for the Master of Science degree in Computer Science at Oklahoma State University in May 1997.

Experience: Worked as a Systems Analyst in the Computer Department of Conoco, Ponca City, OK, from January 1985 to March 1997. Will transfer to the DuPont Advanced Fiber Systems, Richmond, VA, to continue work as a Systems Analyst in April 1997.