Software engineering tools and approaches for Neuroinformatics: the design and implementation of the View-Primitive Data Model framework ('VPDMf')

Gully A. P. C. Burns[1], Fang Bian[2], Wei-Cheng Cheng[2], Shyam Kapadia[2], Cyrus Shahabi[2] and Shahram Ghandeharizadeh[2].

1: Hedco Neuroscience Building, 3614 Watt Way, Los Angeles, CA 90089-2520
2: Henry Salvatori Computer Science Center,University of Southern California, Los Angeles, CA 90089-0781

Abstract

We describe a software-engineering strategy called the 'View-Primitive Data Model framework' (or 'VPDMf') derived from the design of leading commercial software engineering tools. We describe a prototypical implementation of the strategy and its use within neuroinformatics. We present the argument that the only way to fulfill on demands for reliable, easy-to-use software by non-computational communities of neuroscientists is for developers within neuroinformatics to adopt and contribute to approaches such the VPDMf under the open-source paradigm. We present the VPDMf as one such development opportunity.

Keywords

Neuroinformatics, Software engineering, ontologies, data modeling

Introduction

The development of neuroinformatics systems may be accelerated enormously with the use of malleable, cheap, reliable, well-documented tools that could automate the software development process. We describe a computer-aided software engineering (CASE) approach called the 'View-Primitive Data Model framework' (or 'VPDMf'). This system generates a representation of a data model of relational, object-relational or object-oriented systems for subsequent manipulation and use. The core of the VPDMf that captures the design of a data-model is based on the constructs and definitions of the Universal Modeling Language ('the UML', [1]), which is widely becoming a *de-facto* standard for object oriented system-modeling and design. After the data model of the system of interest has been captured, users may define 'views' which act as complex encapsulations of data within the system. Views are constituted by several 'primitives', so that the cardinality of all parts of the primitive is conserved (*i.e.,* if a primitive was derived from data from several relational database tables, the data from each table would have the same number of entries). Views may be linked within a 'vGraph' representation, so that each view may be considered as a node in a graph, where the overlap, containment or associations between primitives defines the graph's edges. These concepts are illustrated in the schematic below:

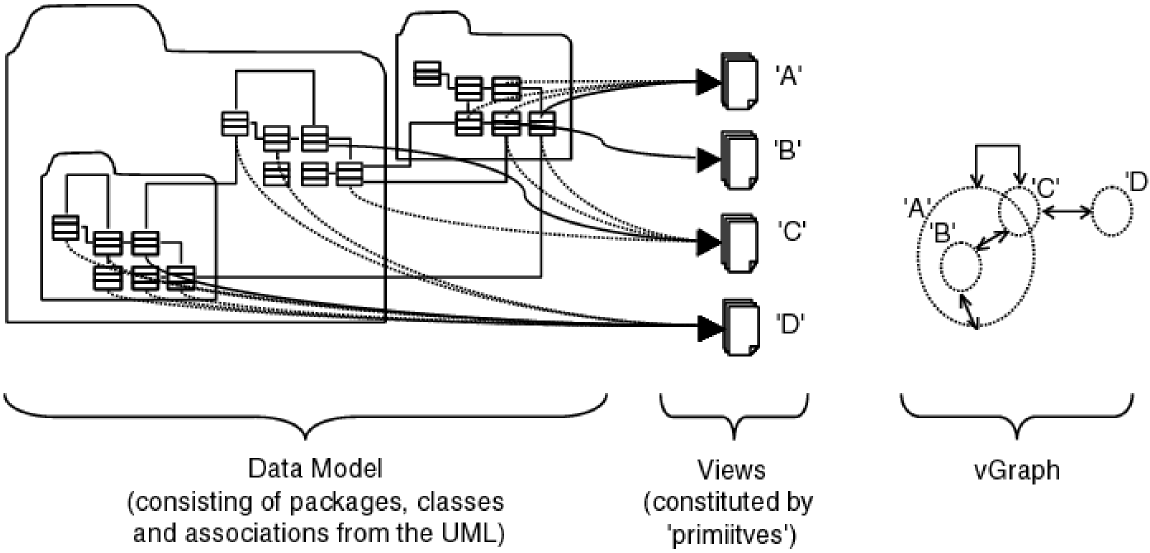


Figure 1: The basic design of the VPDMf. The data model is made up of classes from the UML which are sampled to provide primitives making up views. The overlap and interaction between primitives determines the composition of the vGraph.

The primary purpose of this methodology is to accelerate the development of neuroinformatics systems. This is provided by forward- and reverse-engineering methods, allowing rapid turnaround between the testing, and design phases of project work. An earlier implementation of this system was based entirely on relational databases and was designed to permit data mediation between different systems [2].

Methods

The VPDM framework is implemented as a fully open-source project within the Sourceforge open-source development system. Sourceforge provides a web portal for open-source software development including software for versioning, release-, user-, bug- and documentation-management. At the time of writing (October 2001), Sourceforge administers 29608 projects. The URL for project website is http://vpdmf.sourceforge.net and project page within Sourceforge may be found at http://www.sourceforge.net/projects/vpdmf.

Results

The design of the system's forward- and reverse-engineering capabilities is described in Figure 2. The core set of libraries provides a non-specific representation of the VPDMf (*i.e.,* a data model, views, and vGraphs) that must be populated by loading a data-model from an appropriate source. This source may be a live database system, a database schema, a general data-schema document (such as a Rational Rose Model file), or code for an object-oriented system (such as a Java file). The VPDMf performs translations and data checks to confirm the validity of the data. The user may specify the delineation of views by identifying the constituent elements of each primitive and how primitives link together. This is done in a generic file format that contains no implementation-specific information other that the names of tables, classes or entities comprising primitives.

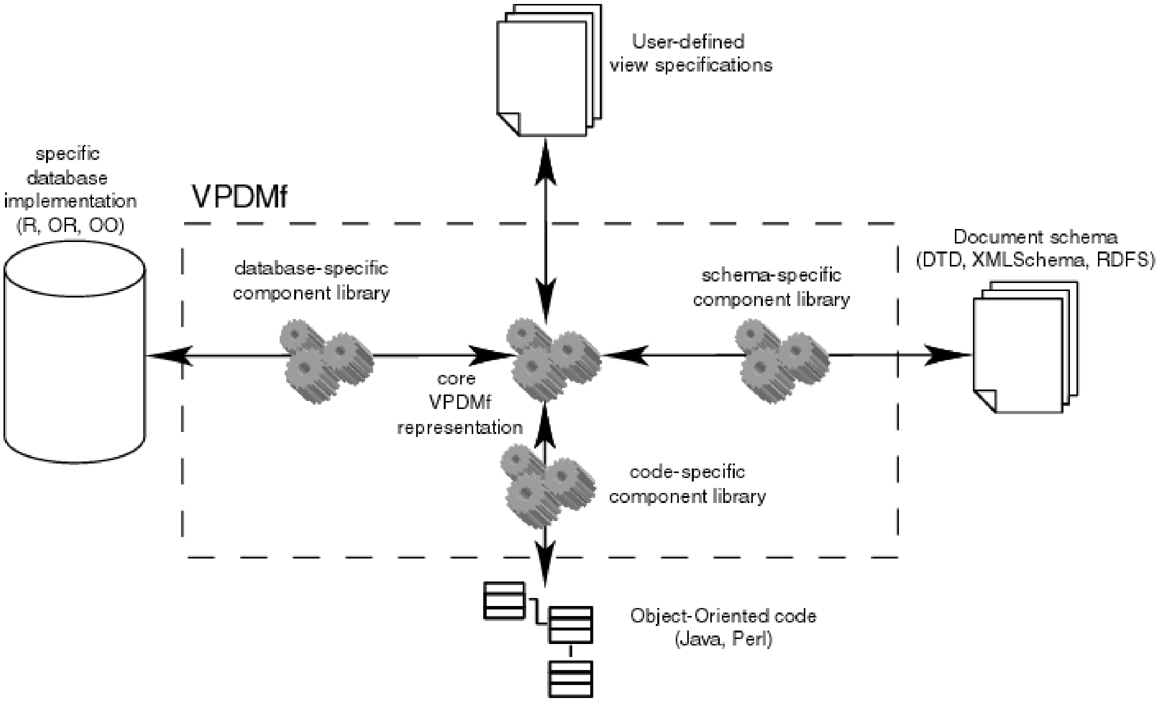


Figure 2: The variety of channels available for schema upload and download within the VPDMf.

Once populated, the core VPDMf system may use different subroutines within the component libraries to generate and execute implementation-specific source code from the generic VPDMf representation. This is a very general approach that has the capability of translating schema from many different types of schema sources to many different types of functional systems. We describe an illustrative case demonstrating forward engineering of a simple UML model from a Rational Rose *.mdl file to a Microsoft Access Database with an web-based user interface, implemented using Active Server Pages (ASPs).

The UML design of the system is expressed as a class diagram in Figure 3. The attributes of the 'publication' class are derived from the so called 'Dublin core' metadata set as a compact representation of the information required to reference a publication as an unambiguous citation [6]). Each publication has an n-to-n association with the 'person' class denoting the paper's authors. Each fragment object contain a datum of information as it appears within the original source material. The 'textual fragment' class is a specialization of the fragment class. Each publication instance may contain several fragment instances.

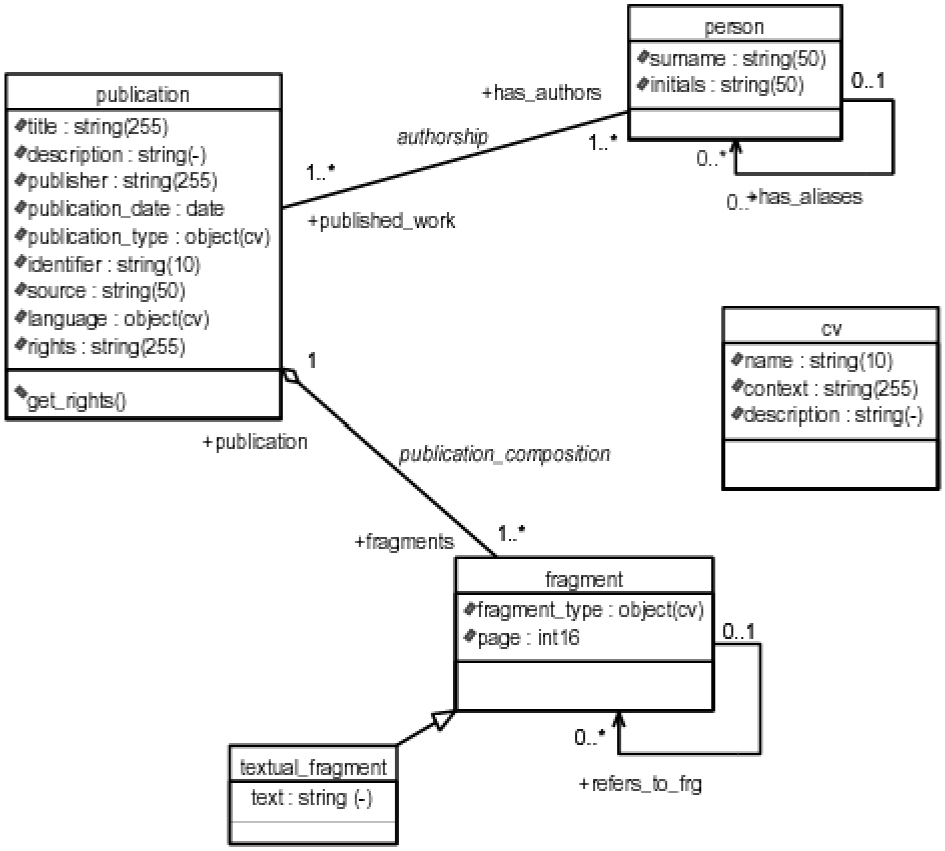


Figure 3: A simple example of a VPDMf schema expressed in the UML, see text for description.

In order to translate this model to a relational schema, we require that each class in the model maps onto a relational table. Each class is given a primary key and additional classes may need to be added to the schema to accommodate the n-to-n relationships within the model. Finally, foreign keys are generated to provide the substrate for references to associations, typed attributes and inheritance relationships. When complete, the VPDMf generates the schema shown in Figure 4.

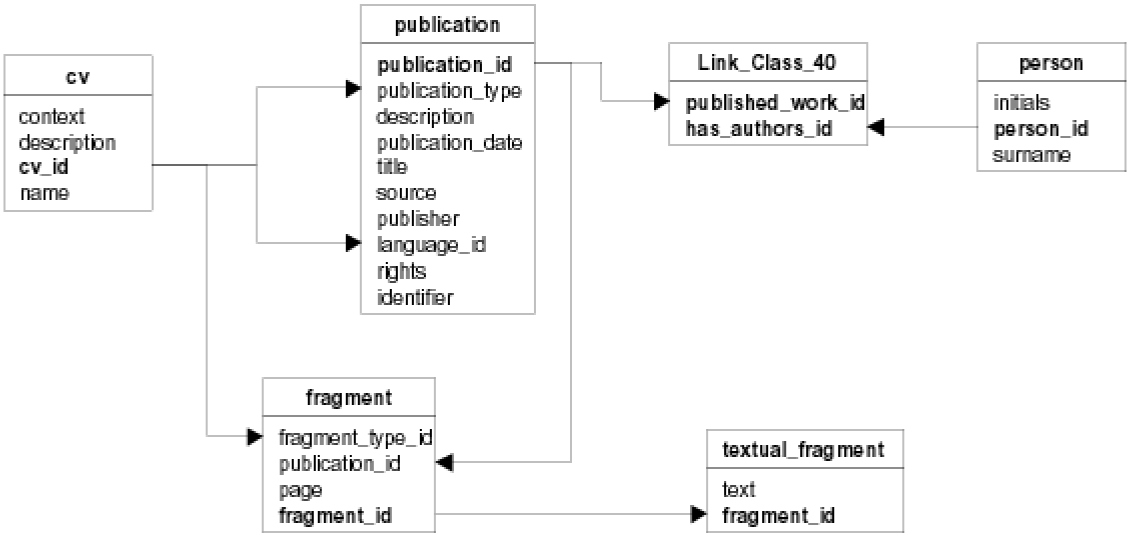
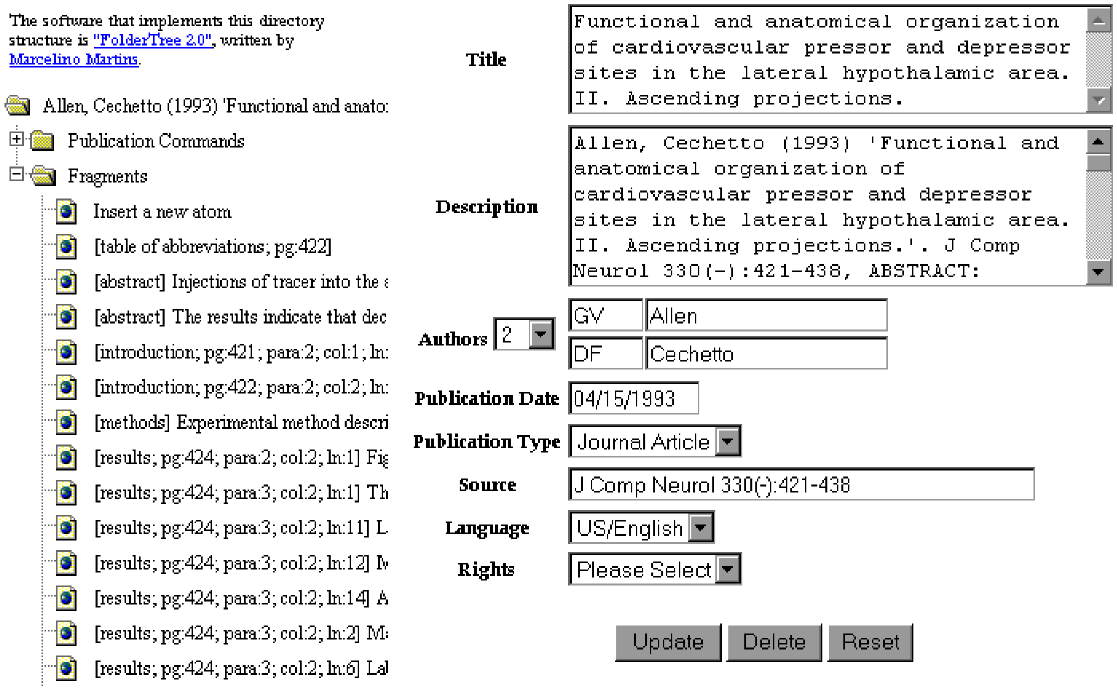


Figure 4: A relational database implementation of the design shown in Figure 3.

In conjunction with the generation of the database, the VPDMf acts to encapsulate the data and then present it to the user in a straightforward and convenient manner. Figure 5 shows an HTML mockup of the user interface of a VPDMf system. The right panel houses a form that can be used to modify data within a typical publication view. Note that data on the line with the 'Authors' label is derived from the person table within the database and has a cardinality of two (i.e., data from two people are represented there). Note how the two primitives (derived from the publication and person tables) appear seamlessly within the same view. The left frame of the page shows a tree-based traversal of the vGraph, illustrating all the fragment views contained within the publication.



The software that implements this directory structure is "FolderTree 2.0", written by Marcelino Martins.

- Allen, Cechetto (1993) 'Functional and anato:
  - Publication Commands
  - Fragments
    - Insert a new atom
    - [table of abbreviations; pg:422]
    - [abstract] Injections of tracer into the ε
    - [abstract] The results indicate that dec
    - [introduction; pg:421; para:2; col:1; ln:
    - [introduction; pg:422; para:2; col:2; ln:
    - [methods] Experimental method descri
    - [results; pg:424; para:2; col:2; ln:1] Fig
    - [results; pg:424; para:3; col:2; ln:1] Th
    - [results; pg:424; para:3; col:2; ln:11] L
    - [results; pg:424; para:3; col:2; ln:12] M
    - [results; pg:424; para:3; col:2; ln:14] A
    - [results; pg:424; para:3; col:2; ln:2] M:
    - [results; pg:424; para:3; col:2; ln:6] Lal

Title Functional and anatomical organization of cardiovascular pressor and depressor sites in the lateral hypothalamic area. II. Ascending projections.

Description Allen, Cechetto (1993) 'Functional and anatomical organization of cardiovascular pressor and depressor sites in the lateral hypothalamic area. II. Ascending projections.'. J Comp Neurol 330(-):421-438, ABSTRACT:

Authors 2 | GV | Allen
DF | Cechetto

Publication Date 04/15/1993

Publication Type Journal Article

Source J Comp Neurol 330(-):421-438

Language US/English

Rights Please Select

Figure 5: Screenshot from an artificial mockup of HTML pages generated by VPDMf. These pages may be provided dynamically.

Discussion

Neuroinformatics is an applied field of computer science, requiring software solutions that impact and enhance the work of non-computational biologists. If the subject is to realize its potential, neuroinformaticians must be able to develop software solutions that operate reliably and understandably, at the heightened level of precision needed to accommodate the academic rigor of scientific research and at a fraction of the cost of commercial solutions.

It is unfortunate that the financial resources generated and required by the software industry are an order of magnitude higher than those generally provided for academic research, leading to the situation where the tools and programming expertise required to build software adequately are not available to the majority of academics. The open-source community provides both programmers and tools at zero cost, permitting the generation of powerful, community-based software that can compete and in some ways surpass commercial products [5]. Within this paper, we present the following argument: firstly, that the open-source methodology is ideal for development work within the discipline of neuroinformatics; secondly: that community-based computer aided software engineering (CASE) systems may provide a uniformity of approach that facilitates the much-desired quality of interoperability between neuroinformatics systems and finally, that the tools described in this paper deliver on both these promises.

Scientists often require an understanding of the inner workings of all parts of their experimental apparatus, including the algorithms they use to analyze and manipulate their data. By its nature, open source software actively encourages users and developers to understand and modify source code at the level of the algorithm. This provides users with unparalleled control over the implementation of software. The work is performed within a distributed development community that is actively committed to enabling users to understand the source code providing expert feedback and help with technical issues.

The core of the VPDMf is based on the constructs and definitions of the Universal Modeling Language ('the UML'), which is widely becoming a *de facto* standard for object oriented system-modeling and design. Although common in industry, broad adoption of CASE software in academia is slow (due mainly to the high cost of these products). An example of a leading academic visual design CASE tool is ArgoUML. It is completely open-source and provides an excellent alternative approach to Rose, but at present has limited forward and reverse engineering capabilities. The Protégé 2000 system provides another alternative with an large user base [4].

One additional important limitation of commercial or large-scale projects is the ease with which they may be adapted by individual users for their own coding purposes. The VPDMf is a fully extensible scripting library that can work in conjunction with existing applications to unify components of a system to a given data model. The system is based on a library of input/output extensions that may be freely written or adapted by users to reverse-engineer or generate source code for their own programs.

References

[1] G. Booch, J. Rumbaugh and I. Jacobson, The Unified Modeling Language user guide.(Reading, MA, Addison-Wesley, 1999).

[2] G.A.P.C. Burns, K. Stephan, B. Ludäscher, A. Gupta, and R. Kötter, Towards a federated neuroscientific knowledge management system using brain atlases, Neurocomputing, 38-40(2001) 1633-1641.

[3] J. Orwant, J. Hietaniemi, and J Macdonald, Mastering Algorithms with Perl, (O'Reilly & Associates, Inc. Sebastopol, 1999).

[4] N. F. Noy, M. Sintek, S. Decker, M. Crubezy, R. W. Fergerson, and M. A. Musen, Creating Semantic Web

Contents with Protege-2000, (IEEE Intelligent Systems 16(2):60-71, 2001).

[5] E. S. Raymond, The cathedral and the bazaar, Musings on Linux and Open Source by an Accidental Revolutionary, (O'Reilly & Associates, Inc. Sebastopol, 1999).

[6] S. Weibel, J. Godby, E. Miller, and R. Daniel, Workshop Position Paper, (OCLC / NCSA Metadata Workshop Report, Dublin Ohio, 1995).

Biosketches

Gully A. P. C. Burns is a research assistant professor at the University of Southern California working as the chief developer of the NeuroScholar system in the laboratory of Larry Swanson. His research is concerned with understanding the large-scale organization of the brain by analyzing patterns of connections between brain structures. This involves theoretical research into databases and data-mining in order to be able to quantify, organize and then analyze data describing the neuronal circuitry in a mathematically tractable way.

Bian, Fang, a Ph.D student at the Departmentment of Computer Science at University of Southern California. Her research interest includes balanced resource allocation in distributed system such as wireless ad hoc network or culter of workstations.

Shyam Kapadia is a second-year graduate student at the University of Southern California completing his masters degree in Computer Science. He is a research assistant and codeveloper of the NeuroScholar system.

Shanshan Song is a graduate student in the Doctoral program at USC.

Weicheng Chen is a graduate student in the Masters program at USC.

Cyrus Shahabi is currently an Assistant Professor and the Director of the Distributed Information Management Laboratory at the Computer Science Department and the Integrated Media Systems Center (IMSC) at the University of Southern California. He has more than fifty articles, book chapters, and conference papers in the areas of databases and multimedia. Dr. Shahabi's current research interests include Multidimensional Databases, Multimedia Servers, and Data Mining. He was the program committee chair of ACM WIDM'99 workshop and is currently serving as a program committee member for ACM Multimedia Conference 2001, VLDB workshop on Technologies for E-Services (TES'2001) and ACM workshop on Multimedia Information Retrieval (MIR'2001).

Shahram Ghandeharizadeh received his Ph.D. degree in computer science from the University of Wisconsin, Madison, in 1990. Since then, he has been on the faculty at the University of Southern California. In 1992, Dr. Ghandeharizadeh received the National Science Foundation Young Investigator's Award for his research on the physical design of parallel database systems. In 1995, he received an award from the School of Engineering at USC in recognition of his research activities. His primary research interests include design and implementation of multimeida storage managers, parallel database systems, and active databases. He has served on the organizing committees of neumerous conferences and was the general co-chair of ACM Multimedia 2000. His activities are supported by several grants from the National Science Foundation, Department of Defense, BMC, Software, and Hewlett-Packard. He is the director of the database laboratory at USC.

Mailing Addresses:

Gully Burns,
Hedco Neuroscience Building,
3614 Watt Way,
Los Angeles,
CA 90089-2520,
gully@usc.edu

Fang Bian,
Henry Salvatori Computer Science Center
University of Southern California
Los Angeles, CA 90089-0781
bian@usc.edu

Shyam Kapadia
Henry Salvatori Computer Science Center
University of Southern California
Los Angeles, CA 90089-0781
kapadia@usc.edu

Shanshan Song
Henry Salvatori Computer Science Center
University of Southern California
Los Angeles, CA 90089-0781
shanshas@usc.edu

Cyrus Shahabi
Henry Salvatori Computer Science Center
University of Southern California
Los Angeles, CA 90089-0781
shahabi@rcf.usc.edu

Shahram Ghandeharizadeh
Henry Salvatori Computer Science Center
University of Southern California
Los Angeles, CA 90089-0781
shahram@pollux.usc.edu