OPEN SOURCE SOFTWARE AS A SCHOLARLY CONTRIBUTION

by

Ali Lotfi
B.Sc., Simon Fraser University, 2003

THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

In the
SCHOOL OF COMPUTING SCIENCE

SIMON FRASER UNIVERSITY

Spring 2009

APPROVAL

Name: **Ali Lotfi**

Degree: **Master of Science**

Title of Thesis: **Open Source Software as a Scholarly Contribution**

Examining Committee:

Dr. Richard T. Vaughan
Assistant Professor of Computing Science
Chair

Dr. Uwe Glässer
Associate Professor of Computing Science
Co-Supervisor

Dr. Arthur (Ted) Kirkpatrick
Associate Professor of Computing Science
Co-Supervisor

Dr. Lou Hafer
Associate Professor of Computing Science
Co-Supervisor

Dr. Toby Donaldson
Lecturer
SFU Examiner

Date Defended/Approved: April_7_2009

Declaration of
Partial Copyright Licence

ABSTRACT

This report addresses the subject of scholarship in open source software. The report presents an expanded model of scholarship due to Boyer and observes that the qualities of scholarship in Boyer's model are exhibited in Open Source software projects. Thirty-two Open Source software projects were examined in the course of researching this report. Analysis of these projects suggests that Open Source software projects can be classified based on three types of contributions: infrastructure for a computing research community, tools for a research community outside computing and robust implementation of important, previously published algorithm. Three representative case studies, Player/Stage (robotics), the Image Processing Toolkit (image processing) and PSAT (numerical computation) are presented in detail in the report to illustrate the presence of attributes of scholarship and the three types of contribution.

Keywords: Open Source Software

Subject Terms: Open Source Software as a Scholarly Contribution

DEDICATION

I dedicate my work to Mohammad Reza Rasmi, my young uncle whom I lost to cancer on September 11th, 2008. Your memory will live with me forever.

ACKNOWLEDGEMENTS

I am mostly grateful to my parents for their encouragement, unconditional love and continuous support in the past four years causing me not to give up and finish my thesis. Of course, this endeavour would not have been possible without the guidance of my dear supervisors. Dr. Uwe Glässer has been a source of inspiration since the time I was in graduate studies and his encouragement to continue and finish my thesis has been instrumental in where I am today. Dr. Arthur Kirkpatrick literally took my understanding of the Open Source environment to another level and without his comments on the numerous revisions of this paper this task would not have been achievable. It is worthy of mentioning that I am forever indebted to Dr. Kirkpatrick's thoughtfulness and understanding during the time that I lost my young uncle to cancer; his encouragement helped me to carry on and finish my thesis. Finally, Dr. Lou Hafer helped me tremendously in the final touches of my thesis and I am truly grateful to have him as one of my supervisors. His genuine smile always reminded me that writing thesis can actually be fun and with positive energy, one can be more efficient.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1: THE TRADITIONAL DEFINITION OF ACADEMIC SCHOLARSHIP

Introduction

This report is organised as an introduction, case studies, and conclusions. A cohesive introduction creates a platform for the discussion of scholarly contribution in Open Source software. The introduction offers an expanded definition of scholarship and explores how the current focus on funded research fits within this model. It also identifies the types of contributions that are found in Open Source software and makes a case for the importance of considering scholarship in Open Source software. In this context, the three case studies are analysed in detail. A conclusion summarises the observations drawn from the case studies.

Motivation

In this section the topic of academic scholarship, which according to [1] is a set of principles and practices employed by scholars in order to make a case for their claim in their respective field, is discussed. This will pave the way for a comparison between the classical research process in academia and the recent research movement in Open Source communities. In the following chapters, Boyer's expanded model of scholarship is presented, portraying the qualities of scholarship in Boyer's model exhibited in Open Source software projects. In this process, the degree of contribution (in Open Source software) is not addressed, but instead the existence of contribution within one or more Boyer categories is shown. I believe such initiative is proposing a revolution in the way we think about classical research and scholarly contribution. Perhaps Feyerabend's Rejection of Universal Rules put it best when he said that "the idea that science can and should be run according to some fixed rules, and that its rationality consists in agreement with such rules, is both unrealistic and vicious" [2]. In the case of this report it would be

unrealistic to not acknowledge the contributions of developers and researchers active in Open Source software platforms which lay the ground work for further research in the respective field. Furthermore, it would be vicious not to recognize the efforts of contributors in an Open Source movement in the context of classical research and scholarly contribution as we attempt to enforce old ideologies and standards.

What is the problem and why is it important?

Denning [3] writes that, during World War II, the U.S. government offered large contracts to selected universities in order to persuade faculty and students to address questions related to war efforts. This initiative was eventually institutionalized with the establishment of the National Science Foundation during the 1950s and reinforced by the creation of the Defense Advanced Research Projects Agency in the early 1960s. It was anticipated that this initiative would benefit the American society as a whole through increased military security, improved public health, and economic prosperity. Over time, the scope of federal support for university research was expanded, so that today we can see that research has become a centrepiece of the university's public identity and faculty are rewarded for their success in attracting research funding.

Denning [4] asks an interesting question, "[...] whether the massive spending on university research produces the value claimed by the Humboldtian and Bushian adherents [pioneers of the federal funding for research in universities]". It seems that something has tarnished the image of university research in spite of the achievements of projects such as the Human Genome Project. Denning points to the "publish or perish syndrome". He notes that in the past half century, almost every university has adopted the standard of granting tenure or dismissing a new faculty member within six years. Further, most junior faculty members are subjected to intense pressure to publish papers in prestigious journals, a routine that many of them continue for the remainder of their academic career. In 1992, about two million scholarly papers in science and engineering were published on an annual basis by approximately 72,000 journals [4]. Denning writes that "The publish or perish syndrome has devalued the original purpose of research in the university." Quantity is often stressed over quality, so that much of this research is mediocre or of no consequence. It is a known fact that the majority of these papers are

read by only a few hundred people at most; worse, more than half of these papers are never cited by another author [4]. It is for this reason that Denning claims that businesses and political leaders are led to doubt whether the anticipated benefits of research to national security, health, and prosperity have actually been realised. Acknowledging the existence of this skepticism, it will be useful to review what constitutes authorship and the qualities which are the hallmarks of scholarship. This will provide a clear base knowledge as we make our case for the importance of recognising scholarship in Open Source software in the remainder of this report.

Academic authorship

Under the topic of academic authorship, publishing journal articles and any original work is the primary vehicle in which academics not only portray the findings of their scholarly work but also gain respect in their respective field. It is no surprise that authorship is the most important metric that is used for assessing the qualifications of a researcher for employment, promotion or tenure. In the process of academic publishing, the authorship usually is claimed by individuals or groups that have made the intellectual contribution for the completion of the research at hand. In the simplest form of academic authorship, the original author is the main engine behind research and publication and most of the time he is the source for further articles or books on the subject at hand. However, it should be noted that the idea of collaborating on a common topic among peers in a respective field is a common practice and at times has been controversial as far as claiming the authorship rights. In order to address such problems, we could potentially define authorship in more broad terms other than just simply writing an article. For example, an individual who is involved in researching a specific topic and through the process comes up with a set of design principles or data related to that research could be considered an author, even if he had a small role creating the final document portraying the results.

What represents authorship?

It is no secret that guidelines and standards of authorship vary between institutions let alone academic disciplines. In order to raise the importance of this subject

and point out the difference in the standards applied in various fields we can point to authorship guidelines in Medicine and Social Sciences in order to gain additional insight into this topic. For example, authorship in the Medical field has a strict set of guidelines. According to the Uniform Requirements for Manuscripts Submitted to Biomedical Journals, in order for a researcher in the Medical field to be considered an author, he should contribute substantially to the concept, design, data collection and analysis of such data as well as drafting the journal article and lastly providing final approval of the version to be published. It is important to note that activities such as obtaining funding or supervising the research group interestingly do not qualify as authorship as the majority of the medical journals essentially have put aside the flexible notion of contributor [5].

Moreover, authorship in the Social Sciences has a somewhat similar set of principles as the Medical field. According to the American Psychological Association (APA), authorship is not defined by the restricted notion of just writing manuscripts but it should encompass those researchers who have been involved in “formulating the problem or hypothesis, structuring the experimental design, organizing and conducting the statistical analysis, interpreting the results, or writing a major portion of the paper” [6].

The development of authorship

If we take a close look at the history of authorship we notice that from the late 1600s to the 1920s the one paper – one author model was the primary vehicle of acknowledgment for the credit that author would receive amongst his peers (7]. However due to the rise of authorship in scientific research, which requires the collaboration of many skilled individuals, we can see that this model has changed a lot since its inception [8].

In this era, it is common to see journal articles in fields such as high-energy physics acknowledging more than 100 authors [9]. Furthermore, we can see the emergence of new models such as Honorary and Ghost authorship, each with its own challenges. Honorary authorship for example is given to individuals who did not have any substantial role in the work presented. The National Academy of Sciences points to the fact that "dilut[ing] the credit due the people who actually did the work, inflate the

credentials of those so 'honored,' and make the proper attribution of credit more difficult" [10].

In the case of Ghost authorship, the opposite scenario happens in which individuals who have made significant contribution to the research at hand are simply not listed as an author. This type of authorship has been connected to collaborations between industry players and academics. It should be noted that Ghost authorship is generally considered to be problematic for the mere fact that it does not address the problems associated with the conflict of interest for researchers involved in the experiment [11].

Academic publishing

When we take a look at the history of academic publishing, we notice that the Philosophical Transactions of the Royal Society of London is one of the earliest research journals which began publication in 1665. It later split into two parts, one for mathematics, physical sciences, and engineering, and one for biology. If we zoom into the 17th century, we can attest to the fact that the process of publishing a research paper was extremely difficult and met great opposition as history has witnessed for Isaac Newton. However, such opposition has significantly been reduced in recent times due the wide acceptance of the process of publishing a scholarly paper in respected academic journals.

In the process of academic publishing, the final product or the contribution of the researcher can be looked at in the context of a paper that could get published in an academic journal. Such paper or article either contains original research results or simply reviews existing results and draws its own conclusions. According to the definition provided by Wikipedia, an Academic Journal “is a peer-reviewed periodical in which scholarship relating to a particular academic discipline is published. Academic journals serve as forums for the introduction and presentation for scrutiny of new research, and the critique of existing research. Content typically takes the form of articles presenting original research, review articles, and book reviews. Academic or professional publications that are not peer-reviewed are usually called professional magazines.”

Moreover, there are some accepted categories of academic papers such as Research paper, Technical paper, System paper, Survey paper, etc., that one can view in the different publishing disciplines such as Sciences, Social Sciences and Humanities. For example, the majority of scientific research papers are initially published in scientific journals and are considered to be the primary source of reasoning for the research material at hand. On the other hand the process of publishing a paper in social sciences can be a very time consuming process as some fields (i.e. economics) require a highly quantitative analysis of the result in order for a research paper to be qualified. Lastly, we can point to the Internet and databases such as Google Scholar, which have provided a new medium for the production and accessibility of academic journals. New movements such as open access in which an author can either self archive or publish her article in a free open access journal have provided a new facility for authors to publish their findings. Acknowledging these new initiatives in academic publishing, the topic of Open Source software as a further extension of scholarship can be discussed. However, before introducing the framework in which we can categorize Open Source software as a scholarly contribution, we must expand the definition of scholarship itself.

CHAPTER 2: EXPANDING THE DEFINITION OF SCHOLARSHIP

Since in this paper we are making a case for Open Source software as a scholarly contribution, we are going to present an expanded model of scholarship due to Boyer and observe that the qualities of scholarship in Boyer's model are exhibited in Open Source software projects. In order to achieve this task, it would be fitting to discuss Boyer's model of scholarship first. Boyer [12] has proposed a model of scholarship with four independent yet interlocking parts: the discovery of knowledge, the integration of knowledge, the application of knowledge, and the scholarship of teaching. The first two types of scholarship, namely the discovery and integration of knowledge, portray the analytical practices of academic life. The third element, namely the application of knowledge, leans more toward the applicability of knowledge to substantial problems. Lastly, scholarship of teaching emphasizes the fact that the work of a scholar "becomes consequential only as it is shared with others" [12].

Boyer's expanded model of scholarship

Ernest Leroy Boyer (September 13, 1928 – December 8, 1995) served as the Chancellor of the State University of New York, United States Commissioner of Education, and President of the Carnegie Foundation for the Advancement of Teaching. Over his lifetime, he was the recipient of over 140 honorary doctorates [13]. Perhaps Boyer is best remembered for his achievements in developing an effective channel of communication between teachers and administrators on how teaching methods and programs should be advanced. He single-handedly changed how education guidelines were viewed during the time he was at the Carnegie Foundation addressing the problems and major issues related to secondary and primary education.

In his book, Scholarship Reconsidered, Boyer states that "it became increasingly clear that one of the most crucial issues – the one that goes to the core of the academic

life – relates to the meaning of scholarship itself" [12]. In his opening chapters, he defines what it means to be a scholar and argues that the time has come to reassess the model of scholarship and reflect on the "variety of functions academics are expected to perform."

Discussing such variety of functions in the academic field requires enlarging one's perspective and Boyer goes to the depth of the issue by addressing the topic of "scholarly". Boyer indicates that in order to be considered a scholar one has to do research and in doing so publication is the primary method of assessing one's scholarly productivity. Boyer points to the fact that because promotion and tenure are highly dependent on the research activities of faculty and further on publishing their results, academics find it "more rewarding" to just deliver a paper. Based on this mindset and incentive, Boyer argues that in order to maintain the high standards of American universities there needs to be a new model of scholarship.

In the discussion of enlarging one's perspective towards scholarly activities, Boyer grabs one's attention by reminding us that the very word "research" itself has just recently entered the "vocabulary of higher education." Moreover, in the discussion of the variety of functions academics are expected to perform, Boyer states that "what we now have is a more restricted view of scholarship, one that limits it to a hierarchy of functions. Basic research has come to be viewed as the first and most essential form of scholarly activity, with other functions flowing from it" [12].

He argues that scholars are academics who perform research, publish and pass on their knowledge to their peers or students and find a way to apply what they have learned or discovered through the course of research. Boyer states that "the latter functions grow out of the scholarship, they are not to be considered a part of it. But knowledge is not necessarily developed in such a linear manner. The arrow of causality can, and frequently does, point in both directions" [12]. He reminds that it is true that "theory leads to practice" but one also should not forget that "practice also leads to theory." He further argues that the process of teaching essentially creates a framework for both research and practice and if one views this topic from this angle, a more comprehensive and flexible model of scholarship can be defined.

In making a case for the broader meaning of scholarship, “one that brings legitimacy to the full scope of academic work” [12], he acknowledges that scholarship in its traditional sense simply means being active in original research material. He further argues that “investigation, looking for connections, building bridges between theory and practice, and communicating one’s knowledge effectively to students” are strong reasons for considering a broader model of scholarship. Specifically, he concludes that “the work of the professoriate might be thought of as having four separate, yet overlapping, functions. These are: the scholarship of discovery, the scholarship of integration, the scholarship of application, and the scholarship of teaching” [12].

Scholarship of Discovery

Under the scholarship of discovery, Boyer argues that “discovery” is closest to the process known as “research” in academia. He states clearly that research is the central pillar of education but quickly reminds his audience that his model “which inquires into the meaning of scholarship, is rooted in the conviction that disciplined, investigative efforts within the academy should be strengthened, not diminished” [12].

He supports the scholarship of discovery by acknowledging the fact that not only is this type of scholarship one of the primary methods of contribution to the existing accessible knowledge of human society as a whole, but also it affects the status and prestige of the university where this process of discovery is conducted.

Scholarship of Integration

Under the scholarship of integration, Boyer “underscore[s] the need for scholars who give meaning to isolated facts, putting them in perspective” [12]. He makes it clear that integration has been discussed in the context of “making connections across the disciplines, placing the specialities in larger context, illuminating data in a revealing way, often educating non-specialists, too” [12]. He does not however suggest backward movement towards earlier time nor does he suggest an amateurish approach, but rather he argues for a concrete and “disciplined work that seeks to interpret, draw together and bring new insight to bear on original research” [12].

It should be noted that the scholarship of integration is closely connected to the scholarship of discovery in the sense that "it involves, first, doing research at the boundaries where fields converge" [12]. Furthermore, Boyer argues that the scholarship of integration "also means interpretation, fitting one's own research – or the research of others – into larger intellectual patterns" [12]. He clearly states that the difference between discovery and integration could be best comprehended by the question at hand. For example, researchers active in discovery ask questions in the line of "what is to be known, what is yet to be found?" and researchers active in integration ask "what do the findings mean? Is it possible to interpret what's been discovered in ways that provide a larger, more comprehensive understanding?" It is these kinds of questions that necessitate the tools of analysis and interpretation.

Scholarship of Application

In the previous two sections we have stated that the scholarship of discovery and integration "reflect the investigative and synthesizing traditions of academic life." Under the scholarship of application, Boyer states that knowledge has a tendency to get more engaged as the scholar asks questions like "how can knowledge be responsibly applied to consequential problems?" [12].

In discussing the scholarship of application, Boyer again reminds us that the term application might be taken out of context "if it suggests that knowledge is first discovered and then applied" [12]. He makes a case for a more dynamic process, one in which new comprehension of knowledge can come out of the application of knowledge itself. In such activities the "theory and practice vitally interact, and one renews the other" [12].

Scholarship of Teaching

Under the scholarship of teaching, Boyer states that "the work of the professor becomes consequential only as it is understood by others" [12]. He makes his point by stating that nowadays "teaching" could be just viewed as a "routine function", but he argues that when teaching is defined as scholarship it persuades future scholars. Being a

master of research in the education field himself he drives his point by quoting Aristotle, who said “Teaching is the highest form of understanding.”

In conclusion, Boyer states that the proposed model of scholarship “divide[s] intellectual functions that are tied inseparably to each other” [12]. Boyer believes that there is tremendous value in recognizing different forms of scholarship in academia “while also acknowledging that they dynamically interact, forming an interdependent whole” [12]. With this framework and vision of scholarship, we could build upon Boyer as we make a case for Open Source software as a scholarly contribution.

CHAPTER 3: OPEN SOURCE SOFTWARE (O.S.S.)

In the 1960's, a process called Request for Comments was employed by researchers with access to the Advanced Research Projects Agency Network (ARPANET) to develop telecommunication network protocols. This movement eventually led to the advent of the Internet in 1969 and can be looked on as one of the first signs of the Open Source movement gaining momentum. The final decision to coin the term "Open Source" came out of a strategy session held in Palo Alto, California, by people involved in the free software movement in response to Netscape's 1998 announcement of releasing its source code. In April 1998, the term "Open Source" was given another big push by the "Open Source Summit", organized by technology publisher Tim O'Reilly. This event brought together the leaders of many Open Source projects such as Jamie Zawinksi of Netscape and after a taking a vote between participants, the Open Source Initiative was formed and the term "Open Source" was adopted.

Open Source Definition

Open Source can be described as a set of standards and adopted methods on how to develop software with the key point that source code is openly available. According to the Open Source Initiative (www.opensource.org) the users of Open Source should be able to get access to and the right to use the code and if the usage is denied the license is categorized as a shared source license which the user is obligated to pay after some trial period. Furthermore there are ten conditions that need to be met in order for source code to be considered as Open Source. Among them are the requirements that the source code should be freely distributable, modifications must be allowed, there be no discrimination against persons or groups to use the code, and other software cannot be restricted. It should be noted that the Open Source Definition is currently used by the Open Source Initiative in order to assess whether a software license can be categorized as Open Source; this definition was created based on the Debian Free Software Guidelines which

were primarily developed by Bruce Perens [14]. Under Perens' definition, Open Source can cover a wide array of software licenses that provide source code available to everyone with no copyright restrictions. Perhaps it is this flexibility that has provided a platform for major commercial users and vendors to create some of the most successful applications in an association of two or more individuals or companies. Some good examples of Open Source products are the Apache HTTP Server and the Linux operating system.

Pros and Cons of OSS

One of the most important advantages of Open Source Software is its inherent ability to target a larger audience in a selected market. Entities involved in the development of Open Source software are often able to create industry standards and thereby gain competitive advantage. According to [14] this factor has caused an increase in developers' loyalty as they feel more empowered and have a sense of ownership of the end product. We can also witness less cost of marketing and logistical services in existing successful Open Source software. Moreover, according to the Harvard Business Review, Open Source Software helps companies to promote their image and their commercial products [15]. There are many other factors that we can associate with the benefit of employing the OSS development approach such as producing reliable, high quality software quickly and rather inexpensively. Supporters of the OSS movement attribute the reliability of OSS to the fact that there are thousands of independent programmers on OSS projects, testing and fixing bugs of the software. They furthermore argue that OSS is more flexible because the modular structure provides a platform for developers to build custom interfaces or simply add new features. According to the Harvard Management update [16] since the final OSS product is the result of collaboration between a large number of programmers, the different perspectives of developers along with their personal goals speeds up innovation.

It should be noted that commercial pressures have historically had a negative effect on the traditional software developer as they have changed their focus from security requirements to customer requirements which are more visible to clients [17].

This pressure is not felt to a high degree in Open Source software as it can be developed in accordance with specific technical requirements and not cave in to commercial pressures. Lastly, we can discuss the motivation of developers for contributing to Open Source projects, which could count as one of the advantages of being involved in the Open Source movement. We can name economic incentives, career concerns and gaining recognition in the software development community as potential motivation factors for the time and energy which developers put into Open Source projects. Authors of [18] argue that developer participation in Open Source projects could be explained by economic theory regarding career concerns. Building on this argument papers such as [19] point to the fact that "greater Open Source participation per se, as measured in contributions made, does not lead to wage increases", however their results also show that working on high profile Open Source projects such as the Apache project indeed leads to higher wages.

As far as disadvantages of OSS, it is believed by some software experts that the Open Source development process might not be well defined and stages such as system testing or documentation maybe ignored. This group of experts list late defect discovery and the lack of empirical evidence showing the quality of Open Source Software as some of their claims [20]. It should be noted that such claims are mostly valid for smaller projects with a low number of developers, as larger successful projects tend to have strict guidelines on testing and documentation of the software, which are enforced by the team of trusted programmers who have earned their way up the chain of development. Case in point, according to GNU (Free Software Foundation) project guidelines for complex projects there is a need for review by few independent developers for even minor changes in order to ensure the quality of software is maintained. Other claims have been made in regards to the inherent weakness of Open Source Software as it relates to its security and providing a platform for hackers to know about potential loopholes of the software. Of course, counter arguments can be made by creating an effective control mechanism for performance of "autonomous agents" who participate in the development of Open Source Software [21].

Open Source Repositories

A source code repository by definition is a place where large amounts of source code are kept. This could be a public or private service which is usually used by multi-developer projects in order to maintain different versions of one project. Repositories provide a platform for developers to submit their bug report, feature request, various patches and comments in an organized fashion. We can name CVS as a popular GNU project that handles these tasks for Open Source projects.

In our research to date we have found a list of twenty-six repositories for a variety of software development languages and Operating System platforms. We certainly did not expect the list of repositories to be this extensive but, as we will name a few repositories below one can see a clear organized movement towards keeping track of each community's contribution along the way for specific projects.

The largest Open Source Software repository is called SourceForge.net which provides free hosting to Open Source Software development projects with a centralized resource for managing projects bugs, communication and code. It is interesting to note that SourceForge was originally itself Open Source software which was commercialized at version 2.5. SourceForge.net as a frontier of repositories is a collaborative revision control and software development management system. It not only provides the front-end of software development lifecycle services but it also integrates itself with a number of Open Source applications such as PostgreSQL and Subversion. For each Software hosted on SourceForge the community surrounding the code base can benefit from features such as customized mailing lists for both developers and users, detailed statistics for bugs reported, feature requests, support requests, number of downloads, user activities, access to previous versions of the software as well as up to date announcements regarding the status of the software.

For the benefit of other researchers, in the appendix section, we have provided a list of twenty-six repositories that we have found in the course of our research for this thesis. Each repository has a short description along with a URL to its official website.

CHAPTER 4: DEMARCATING SCHOLARLY OPEN SOURCE SOFTWARE

The accepted scholarly contribution protocol in computing science suggests that publishing papers on the ideas embodied in specific software projects is a more significant contribution than the software itself. This mindset in essence does not provide the necessary incentive for Open Source software in making a scholarly contribution through writing codes. With reference to the above definition of scholarship presented by Boyer, we will show that attributes of Open Source software match attributes of the expanded definition of scholarship. For instance, scholarship of discovery can be looked at in the context of proposing a new algorithm to address a specific problem and further on under the scholarship of integration of knowledge making sure that this algorithm is integrated in usable source code format to be applied in a specific field of study. Under the scholarship of teaching, the Open Source movement by definition provides the means for other developers to participate in the development of a more cohesive code base causing the transfer of knowledge and further research in the respective field.

In the course of conducting research for this report, thirty-two Open Source software projects were examined as potential candidates for case studies. In this thesis we are making a proposal that each of the chosen case studies can be classified based on three types of contributions. In the detailed analysis of each case study, I will show the relationship between these three types of contributions and Boyer's proposed model of scholarship. Chapter 2 expanded the notion of scholarship beyond publishing articles in specialist journals and this chapter proposes criteria for determining whether an open source project is intended to be evaluated as a scholarly contribution. It is the goal of this section to provide demarcation criteria and establish a clear framework for OSS projects that make a scholarly contribution.

Proposed types of contributions

The question that we came up with when we started this research was “Can OSS itself be a scholarly contribution beyond the ideas it implements?” Our answer to this question was a resounding “Yes”. We propose three types of contributions made by Open Source software projects and through discussing these categories along with their corresponding case studies, we will make a case for this claim throughout this paper. The three proposed categories are:

1. Infrastructure for a computing research community: software that provides infrastructure for further research in the area addressed by the software.
2. Tools for a research community outside computing: applying known Computing Science techniques to further research in another field.
3. Robust implementation of important, previously published algorithms: implementation of an algorithm that was described in a paper but not implemented, or where the only prior implementations were proprietary or brittle proof-of-concepts.

Case studies

We have chosen a candidate for each category of our framework:

- Player/Stage (robotics) representing “Infrastructure for a computing research community”
- Image Processing Toolkit (image processing) representing “Tools for a research community outside computing”
- PSAT (numerical computation) representing the “Robust implementation of important, previously published algorithms”

Before outlining a detailed list of thirty-two case studies that were considered for this thesis along with their corresponding category of contribution, it is important to discuss the mindset behind choosing the above three case studies over the rest. First off, in the case of Player/Stage we had access to an Open Source software base with an established

track record, active community surrounding it and ongoing research papers that are based on the original work of the authors of Player/Stage. Such factors, specifically the fact that Player/Stage has facilitated further research (as we will show in its corresponding section) were the reason for choosing it as a candidate of "infrastructure for a computing research community". It should be mentioned that both ITK and PSAT demonstrated the same characteristics as Player/Stage providing us with research material necessary to prove our case that Open Source software can make a scholarly contribution based on the three categories that we have proposed here.

Secondly, because we are showing the presence of scholarship in reference to Boyer's model, our demarcation criteria focuses on candidates that have specifically facilitated further research. For example in the case of Open Office, despite the fact that we are dealing with an Open Source software project which improves productivity we could not claim that this OSS has facilitated further research. Therefore, our proposed categories of contribution have a scholarly mindset; hence, we have included a research section in the discussion of each case study showing the importance of the software in further nourishing research that was not possible prior to its development.

Infrastructure for a computing research community

There were fifteen candidates proposed for the "infrastructure for a computing research community" category:

1. **Player/Stage**: The Player Project creates Free Software that enables research in robot and sensor systems. The Player robot server is probably the most widely used robot control interface in the world. Its simulation backends, Stage and Gazebo, are also very widely used. http://playerstage.sourceforge.net/
2. **OpenSolaris**: The OpenSolaris project is an Open Source community and a place for collaboration and conversation around OpenSolaris technology.
3. **Simics**: Virtutech® Simics™ is a full system simulation platform, capable of simulating high-end target systems with sufficient fidelity and speed to boot and run operating systems and commercial workloads.

4. **SimpleScalar**: SimpleScalar tools use DEF files to specify the intricacies of instruction semantics, dependencies and disassembly.
5. **The Xen VMM** (virtual machine monitor) is an Open Source project that enables users to create many virtual machines, each of which runs an instance of an operating system.
6. **Linux** is a free Unix-type operating system originally created by Linus Torvalds with the assistance of developers around the world.
7. **L4** is the name of a second-generation µ-kernel (microkernel) designed and implemented by Jochen Liedtke, running on i486 and Pentium CPUs.
8. **Transactional memory (TM)** simplifies parallel programming by guaranteeing that transactions appear to execute atomically and in isolation.
9. **SUIF compiler infrastructure**: aims to help programmers to write more reliable programs, detect errors in their programs and diagnose errors.
10. **TinySTM** is a lightweight and efficient word-based STM (Software Transactional Memory) implementation. Its time-based algorithm is derived from LSA (Lazy Snapshot Algorithm) and its lock-based design borrows several key elements from other word-based STMs, such as TL2. http://tinystm.org/
11. **u8u16**: The software demonstrates high-speed encoding form conversion several times faster than typical industry standard iconv implementations. http://u8u16.costar.sfu.ca/
12. **XML::Lite** is a lightweight XML parser, with basic element traversing methods. It is entirely self-contained, pure Perl (i.e. not based on expat). http://aspn.activestate.com/ASPN/CodeDoc/XML-Lite/Lite.html
13. **SOAP::Lite** for Perl is a collection of Perl modules, which provides a simple and lightweight interface to the Simple Object Access Protocol (SOAP, also known as Service Oriented Access Protocol) both on client and server side. http://www.soaplite.com/

14. **vuVolume** is a framework for volume rendering algorithms in C++. When creating a new rendering method in vuVolume it is possible to just concentrate on the actual algorithm and let the framework take care of the more administrative tasks. http://www.files-library.com/files/vuVolume.html
15. **lmbench** is a suite of simple, portable, ANSI/C microbenchmarks for UNIX/POSIX. In general, it measures two key features: latency and bandwidth. lmbench is intended to give system developers insight into basic costs of key operations. http://www.bitmover.com/lmbench/

Tools for a research community outside computing

There were four candidates proposed for the "tools for a research community outside computing" category:

1. **ITK** (Insight Segmentation and Registration Toolkit), ITK is an Open Source software system to support the Visible Human Project. http://www.itk.org
2. **TeX**: together with the METAFONT language for font description and the Computer Modern typeface, it was designed with two main goals in mind: to allow anybody to produce high-quality books using a reasonable amount of effort, and to provide a system that would give the exact same results on all computers, now and in the future.
3. **Stanford GraphBase**: The Stanford GraphBase is a collection of programs and datasets which generate and manipulate graphs and networks.
4. **A learning kit** is a collection of digital learning materials (a learning object) and a software application called gStudy. gStudy supports learners' interactive engagement with multimedia information in the learning kit to learn, apply and transfer that information to new situations. http://www.learningkit.sfu.ca/

Robust implementation of important, previously published algorithm

There were 12 candidates proposed for the "robust implementation of important, previously published algorithm" category:

1. **GNU Octave (PSAT)** is a high-level language, primarily intended for numerical computations. It provides a convenient command line interface for solving linear and nonlinear problems numerically, and for performing other numerical experiments using a language that is mostly compatible with Matlab. http://www.gnu.org/software/octave/
2. **Vision**: The Open Computer Vision Library is a collection of algorithms and sample code for various computer vision problems. The library is compatible with IPL and utilizes Intel Integrated Performance Primitives for better performance: http://sourceforge.net/projects/opencvlibrary/
3. **SPLASH**: Stanford Parallel Applications for Shared Memory (SPLASH) http://www-flash.stanford.edu/apps/SPLASH/
4. **Parabix**: (Parallel bit streams for XML - TM) demonstrates a fundamentally new way to perform high-speed parsing of XML documents. http://parabix.costar.sfu.ca/
5. **PROMISE**: is a novel peer-to-peer media streaming system encompassing the key functions of peer lookup, peer-based aggregated streaming, and dynamic adaptations to network and peer conditions. http://promise.costar.sfu.ca/
6. **P2pcache**: Peer-to-peer (P2P) file sharing systems generate a major portion of the Internet traffic, and this portion is expected to increase in the future. http://p2pcache.costar.sfu.ca/
7. **FFTW**: is a C subroutine library for computing the discrete Fourier transform (DFT) in one or more dimensions, of arbitrary input size, and of both real and complex data http://www.fftw.org/
8. **Gnuplot**: is a portable command-line driven interactive data and function plotting utility for UNIX, IBM OS/2, MS Windows, DOS, Macintosh, VMS, Atari and many other platforms. http://www.gnuplot.info/
9. **Pin**: is a tool for the dynamic instrumentation of programs. It supports Linux binary executables for Intel (R) Xscale http://rogue.colorado.edu/pin/

10. **Intel® VTune™**: Performance Analyzer makes application performance tuning easier with a graphical user interface and no recompiles required. http://www.intel.com/cd/software/products/asmo-na/eng/vtune/239144.htm
11. **Gprof**: Profiling allows users to learn where their program spent its time and which functions called which other functions while it was executing. http://sources.redhat.com/binutils
12. **CacheTrack**: An adaptive hierarchical cache management system for improving effective cache hit ratios by eliminating unnecessary duplicate cache entries in two coupled cache memories. http://www.patentstorm.us/patents/5627990.html

Example of an Open Source project that is not scholarly

OpenOffice (OO.o or OOo) is an office application suite available for a number of different computer operating systems. It supports the OpenDocument standard for data interchange as its default file format, as well as Microsoft Office '97-2003 formats, among many others (http://www.openoffice.org/). We have chosen OpenOffice as an example of an Open Source project that is not a scholarly contribution, since it fails our demarcation criteria due to not facilitating further research.

CHAPTER 5: PLAYER/STAGE

Player/Stage is one of the most widely used robot control interfaces in the world, which is currently being employed in more than 25 countries and used by entities such as Boeing Company, the Australian Centre for Field Robotics and Chinese National University of Defense Technology [22]. After discussing Player/Stage's technical information, in a subsection of "User Community" we will objectively show how a research community has formed around the original idea of Player/Stage and how this original idea has been extended by academic researchers to this date is being referenced to creating a platform for research in robot and sensor systems. Lastly, a series of research papers along with concluding remarks are discussed portraying the existence of a contribution (infrastructure for a computing research community) and the connection to Boyer's model of scholarship.

Technical information

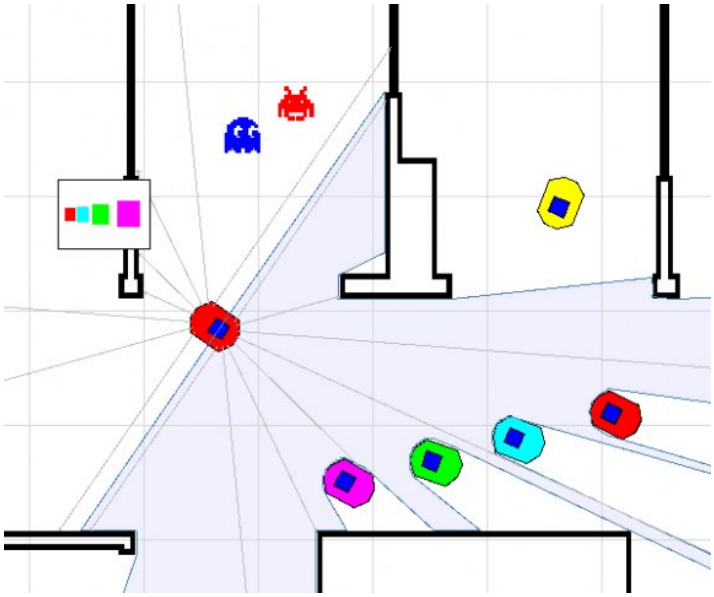
Player Project

The purpose of the Player Project is to facilitate developing Free Software, which in turn helps research in robot and sensor systems. The Player robot server, along with its simulation backends Stage and Gazebo, is one of the most widely deployed robot control interfaces in the world. In simple terms, Player supplies a network interface to a mixture of robot and sensor hardware. The practical aspect of Player/Stage design lies in the client/server model, which enables programmers using different languages to run a Player program on any computer with a network connection to the robot. The client/server model supports multiple simultaneous client connections to machines, thereby facilitating a platform for distributed sensing and control.

Stage Project

Stage replicates a group of mobile robots moving in and sensing a two-dimensional bitmapped surrounding. As part of this project, multiple sensor models are developed including sonar, scanning, laser rangefinder, pan-tilt-zoom camera with colour blob detection and odometry. Stage is normally employed as a Player Plugin module acting as a "lightweight, highly configurable robot simulator that supports large populations" [24]. The way it works is to have the user write robot controllers and sensor algorithms in the form of 'clients' to the Player 'server'.

Figure 5.1: Stage 2D robot simulator.



Screen shot of Stage 2D robot simulator (v2.0.0), simulating a population of mobile robots, sensors and objects in a two-dimensional bitmapped environment. © 2008, Richard Vaughan, by permission.

User Community

Developers

In the official website of Player/Stage (http://playerstage.sourceforge.net) maintained by pioneers of this infrastructure platform, one quickly recognizes that indeed

a focused community has been created around the base idea that was started in early 2000. This fact is easily supported by the number of active researchers who are contributing to this idea continuously either through writing code or simply researching and bringing new ideas based on the original platform as we show in the following section.

As of May 26, 2008 there are 12 main developers who have access to the Player Project Subversion repository. For each person, original authors have supplied SourceForge.net username, a specific role, and detailed the directories to which each developer is allowed to commit. Because SourceForge.net's Subversion setup does not allow per-directory access control, then it would be up to the developers and maintainers to pursue and impose this policy.

Also, the GNU General Public License (GPLv2) permits anyone to make use of and share the source code of Player/Stage. This fact makes it very difficult to acquire the exact number of Player users. The pioneers of Player/Stage believe that that there is no reliable mechanism for user registration and reporting and perhaps the best trusted sign of showing that a platform such as Player/Stage is used and extended is the number of developers, contributors and entities worldwide making contribution to Player/Stage infrastructure base.

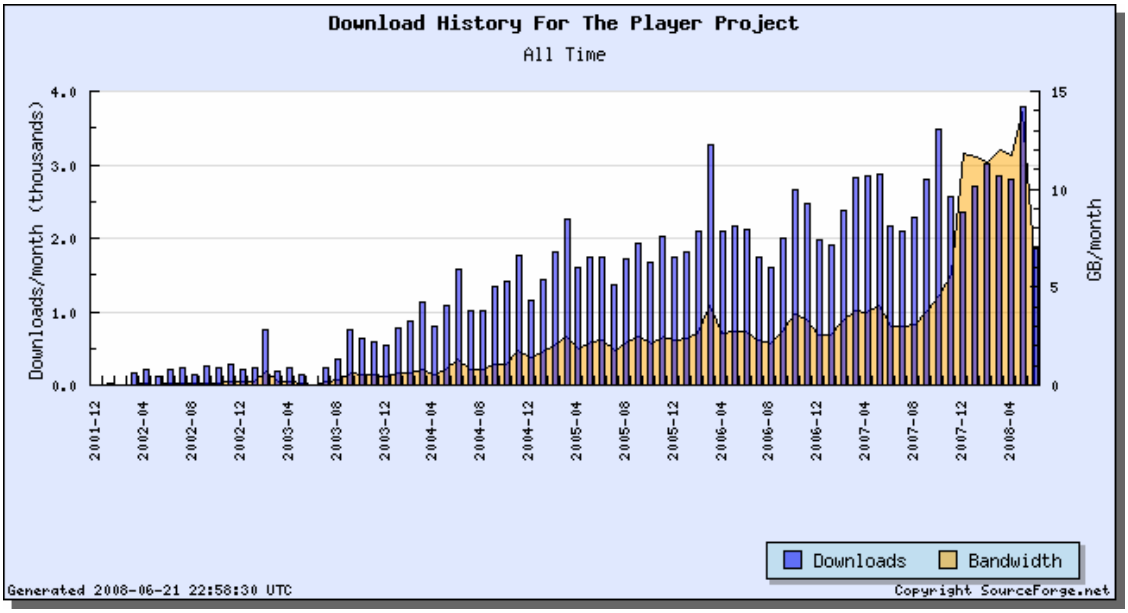
Source Forge statistics

After taking an initial look at SourceForge.net we can see that we have a few sources of statistics at our disposal in order to analyze the usage of Player/Stage. For example, we have access to download statistics, support forum messages and bug tracker items, submissions to the laboratory maintained by authors and most importantly references in published robotics articles, which we will address in the next section. Authors have also used Google’s web analytic to show the diverse set of users that download this software from the four corners of the world. As of June 7th 2008, there are four project administrators and twelve developers active on Player/Stage. Administrators believe that the intended audiences of Player/Stage are researchers in Education and

Science; therefore, they have specifically categorized Player/Stage under Education, Artificial Intelligence and Robotics.

The first interesting Figure 5.2 that we can analyze portrays the Download History Statistics for the Player Project from 2001 to 2008. One can immediately notice that there are a few jumps in the number of downloads in years 2002, 2004, 2006 and an exceptional increase in bandwidth in 2007. In June of 2002, the long-awaited Player/Stage 1.2 is released which justifies the spike that we see in the graph below. This is further supported by the fact that July 2002 had the most number of downloads (240) and bytes served (125.8 MB) in all of 2002. This pattern holds true as we see the first release of Stage after a major rewrite in late 2004 causing the spike in graph with the most number of downloads (1773) and bytes served (1.7 GB) in all of 2004. In early 2006 Player 2.0.0 and Stage 2.0.0 are released causing the most number of downloads (3270) and bytes served (4.1 GB) in all of 2006 supporting the fact that live community is indeed interested and growing around Player/Stage. The tremendous increase in bandwidth in late 2007 early 2008 can only be attributed to the increase in size of the packages downloaded by users as the number of downloads did not drastically change compare to 2007.

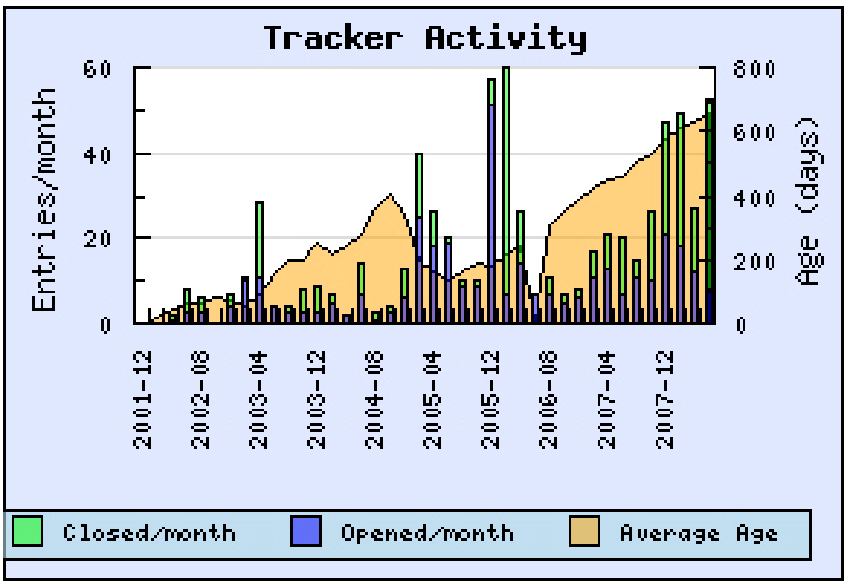
Figure 5.2: Statistics for Player Project



Download History Statistics for Player Project from 2001 - 2008

The second graph Figure 5.3 that can provide us with insight on activities of developers and users of Player/Stage on SourceForge.net is the Tracker Activity graph, which follows items such as number of patches, support and feature requests since 2001. There are few spikes in the graph in years 2003, 2005 and 2007 that demand further analysis. For instance in 2003 there were a total of 70 entries opened, out of which 63 were closed by the end of the year. This could be explained by the fact that Player/Stage 1.3 was released towards the end of 2002 causing a surge of activity in 2003. Again, we see a spike at the beginning of 2005 with a total of 215 entries opened, out of which 83 were closed by the end of the year. This could be correlated with the fact that in late 2004, Stage 1.6 was released after a major rewrite causing a surge of activity in feature requests and bug reports in the community. Furthermore, we can link the release of Player/Stage 2.0.3 in late 2006 to the peak up in activity in 2007 in which a total of 116 entries were opened and 133 were closed by the end of the year.

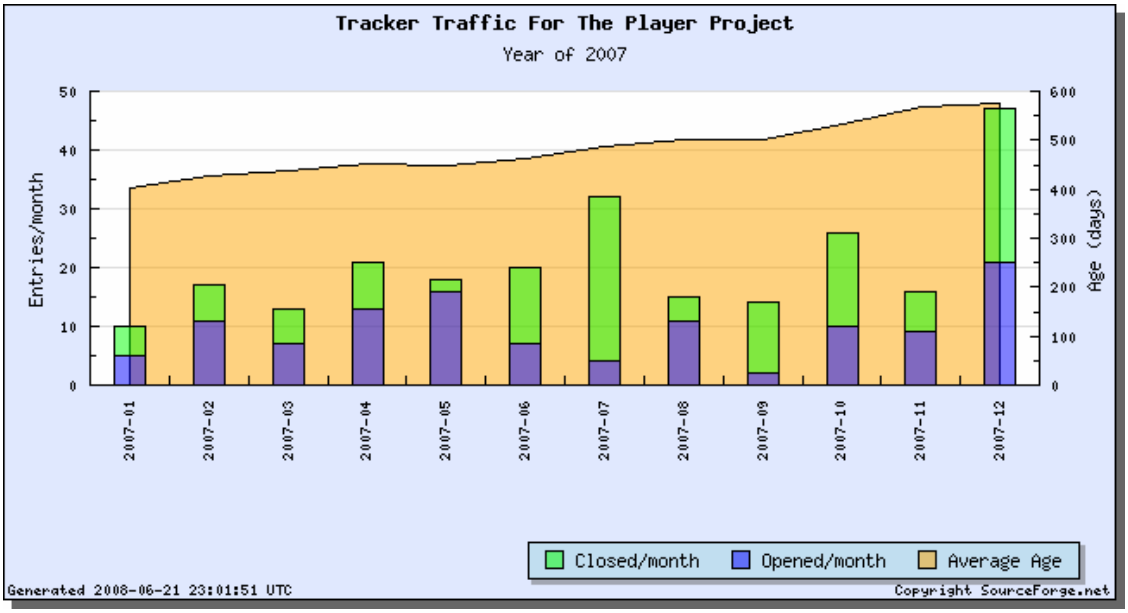
Figure 5.3: Tracker Activity Statistics



Tracker Activity Statistics for Player Project from 2001 – 2007

Moreover, our claim that Player/Stage is a growing and dynamic community can be supported by analyzing the four mailing lists designated to Player/Stage, CVS repositories and SVN repositories. There are four mailing lists provided via a SourceForge.net version of GNUMailman to address communication needs between members of the community. The first mailing list is designed for Player/Stage commit archive with 3246 messages and a total of 20 subscribers. For instance in October 2007 the number of posts peaked to 298 showing a sign of increase in the activity of subscribers due to a new version release for Stage, specifically version2.1.0rc1 in which the number of downloads peaked to 1901. As we can see in the following graph, the number of opened and closed activities has surged right after the end of October which is consistent with the increase in the number of posts in Player/Stage commit archive mailing list.

Figure 5.4: Tracker for Player Project



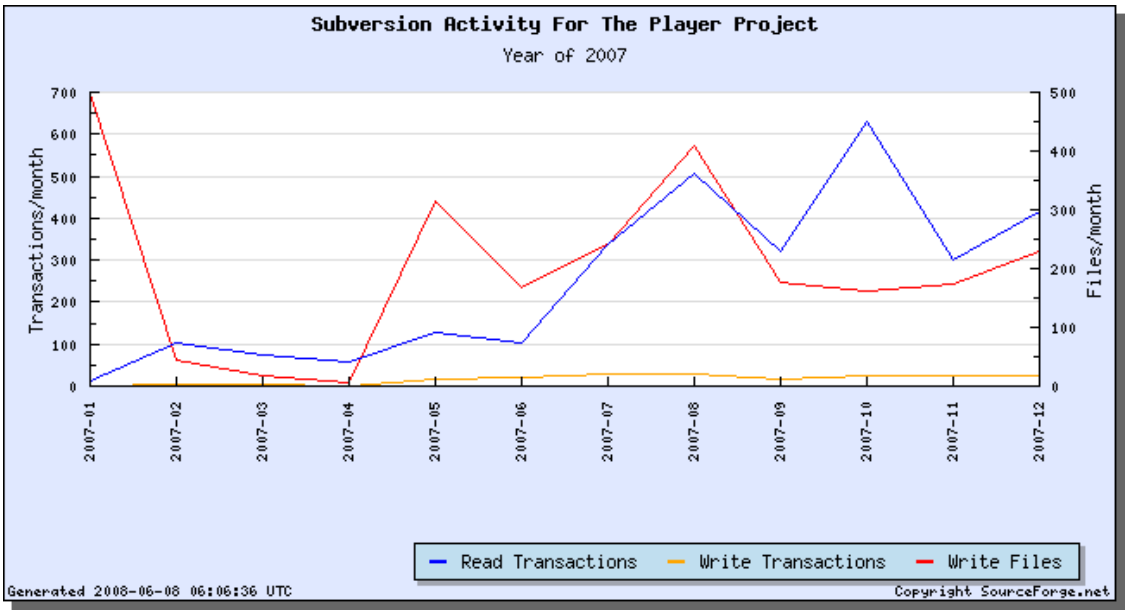
Tracker Activity Statistics for Player Project for year 2007

The flow of activity in the second mailing list which is designed for developers (3075 messages and 200 active subscribers) and the third mailing list which is designed for users of Gazebo (2785 archive messages with 168 subscribers) is consistent with the Tracker Activity graph which we discussed above. This fact is furthermore supported by correlating the data received from the fourth mailing list designed to track Player/Stage users (7710 archived messages and 399 subscribers) with the full graph of Tracker Activity and Download History Statistics analyzed above. For instance after going through the mailing list it is evident that there are four major surges of communication in April of 2005 (189 posts), April of 2006 (209 posts), August of 2007 (204 posts) and April of 2008 (249 posts). If we take a close look at the Download Statistics graph we can see that indeed not only the number of downloads have increased in these specific months but also the number of Closed/Open features or bug requests have increased specifically in the same months according to the Tracker Activity graph.

The last section that we would like to discuss is related to CVS and SVN Repository activities keeping in mind that SVN was chosen as a primary repository in 2008. As of June 8th 2008, there were 720 commits, 84,831 reads in the CVS Repository

and 564 commits, 19,589 reads in the SVN Repository. For instance, Subversion Activity graph for year 2007 is displayed below, showing fluctuations throughout 2007 with peaks in May, August and October of 2007. These increases in activity are directly correlated with the representation of data that we receive in Download History and Tracker Activity Graph.

Figure 5.5: Subversion for Player project



Subversion Activity for Player Project – 2007

Related projects

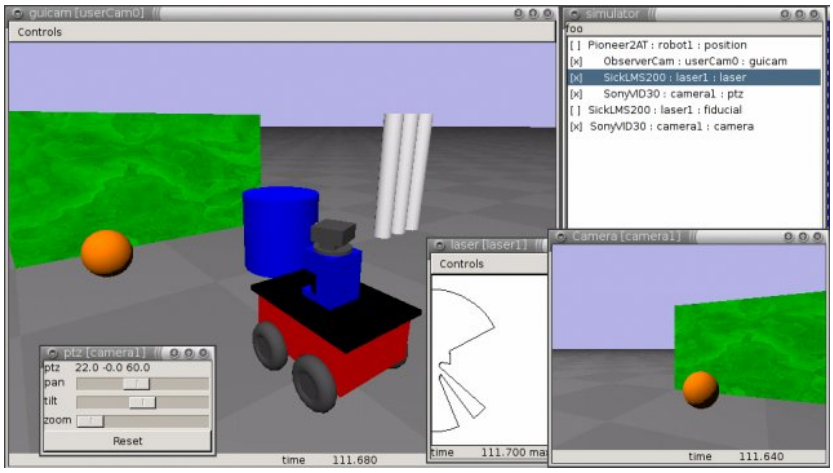
Gazebo 3D multiple robot simulator

Gazebo like Stage has the ability to simulate a group of mobile robots but does so in a three dimensional world. One of its key elements is to provide a practical sensor reaction as well as reasonable communications between objects.

As we have stated above, Player/Stage project provides two multi-robot simulators: Stage and Gazebo. Since both of these simulators are well-matched with Player, client programs can run on both of these simulators with little or no change of

source code. The key distinctions between these two simulators lies in the fact that Stage is methodically developed to simulate large numbers of robots with a low degree of fidelity while Gazebo is developed to simulate small numbers of robots with a high degree of fidelity. This fact makes the two simulators match very well, enabling users to alternate if there is a need.

Figure 5.6: Gazebo 3D robot simulator (v0.5.1).



Screenshot of Gazebo, a multi-robot simulator for outdoor environments. Like Stage, it is capable of simulating a population of robots, sensors and objects, but does so in a three-dimensional world. © 2008, Nate Koenig, by permission.

Research articles referencing Player/Stage

The modular design of Player/Stage has caused the robotics community to hail the arrival of the Player/Stage project. Given the relative small size of the robotic community worldwide, one can say that since Player/Stage is used in more than twenty major academic and industrial research labs around the world as well as being taught in graduate and undergraduate classes, its momentum on a global scale has been a success story [24]. As we discussed in Chapter 2, Boyer's scholarship of teaching emphasizes the fact that the work of a scholar becomes consequential only as it is shared with others. The rapid success of Player/Stage would have not been possible if the original authors of Player/Stage did not have the lofty goals of making this project Open Source and hence

sharing it with the research community. This initiative by itself is the first scholarly contribution directly related to scholarship of teaching presented by Boyer as the work of scholar “becomes consequential only as it is shared with others” [25]. Furthermore, in order to explore in detail the candidacy of Player/Stage as an “infrastructure for a computing research community” and its relationship with Boyer’s scholarship model nine research papers have been considered.

Research Paper 1

The first paper [26] of interest, which was published in IROS and makes reference to Player/Stage, presents a multi-modal object attention system that is able to identify objects referenced by the user with gestures and verbal instructions. For instance, the robot’s hardware is guided by the Player/Stage software “providing an interface to the robot’s sensors and actuators” [24]. This facility enabled the authors to substitute the controller with a more sophisticated piece having features such as obstacle avoidance. This enhancement by way of employing Player/Stage acknowledges the existence of Boyer’s integration of knowledge. Moreover, integration of knowledge from Player/Stage has led to the discovery of new knowledge namely the ability to identify “objects referenced by the user with gestures and verbal instructions”. Such discovery could qualify under Boyer’s scholarship of discovery. It is important to note that none of the four authors were part of the Player/Stage development team, pointing to the fact that the research community has built on the original idea of Player/Stage and it is using it as a base for further research. This paper [26] has been cited by 25 other articles mostly in the Robotics field according to Google Scholar.

Research Paper 2

The second publication [27] of interest, which employs Player/Stage robot control software libraries presents a multiple target tracking approach for following and passing persons in the context of human-robot interaction. For instance in order to communicate with sensors and the robot controller, the authors use Player/Stage robot control software libraries. The authors of [27] clearly explain that the control module manages “incoming

sensor information and sends it to the respective modules for interpretation". The integration of Player/Stage libraries for this task acknowledges the existence of Boyer's scholarship of integration.

Moreover, they show how the resulting commands for motors are produced and passed on to the robot controller via the interface of the Player robot control server. Again, integration of the Player robot control server is a sign of the existence of a contribution as it relates to Boyer's scholarship of integration. According to Google Scholar this article has been cited 23 times since 2005 by articles which have appeared in IEEE, ACM SIGCHI/SIGART Human-Robot Interaction and Control Mechanisms for Spatial Knowledge Processing conference to name a few, a testament to the fact that a real research community is finding value in the work that has used Player/Stage. The last note is to point that neither of the two authors were part of the development team of Player/Stage who essentially saw an opportunity to do further research based on the infrastructure provided by Player/Stage.

Research Paper 3

The third paper [28] that we would like to discuss in lights of using Player/Stage as a base, addresses the problem of tracking multiple targets by means of a network of communicating robots and stationary sensors. The authors practically discuss a Region-based strategy, which in turn controls robot operation at two levels. The usage of Player/Stage is clearly visible throughout this project as the same Player interface is used with both physical and simulated devices acknowledging the existence of Boyer's scholarship of integration; the tracking system in turn was created in Stage and was easily transitioned to real robots.

The main issue addressed in this paper is to create an "on-line, coordinated, motion strategy for robot positioning" [28]. In order to evaluate their approach the authors conducted test cases with ActiveMedia and Player/Stage Software Platform. Reasoning that Player inherently provides the "same interface to real Pioneer robots" [28], the authors decided to employ the same specification for real-robot simulations. As of June 6th 2008, this paper has been cited by 52 other publications, according to Google

Scholar, most importantly noting that none of the authors were part of the team of developers active on Player/Stage.

Research Paper 4

The fourth paper [29] that uses Player/Stage, targets resource allocation from a learning point of view with the goal of creating an adaptive characteristic in robots and multi-robot systems. This project presents experiments, in simulation, with a group of robots that improve their performance on a simple task by using support learning to link input states with a set of abstract behaviours. Interestingly we can see the deployment of Stage to run simulation trials faster than real time acknowledging the existence of Boyer's scholarship of integration. Such improvement has produced considerable amounts of data needed to decide the run-time behaviours and could qualify under Boyer's scholarship of discovery as it relates to improving resource allocation.

Research Paper 5

In [30], "experiments were carried out using the Stage simulator and Player devices" addressing large-scale human system interaction. In discussing large-scale human system interaction, the authors present an interaction infrastructure in line with providing two phases that accommodate many-to-many interactions and generalized one-to-one interaction between users and several systems in simulation. "Player provides a suite of software clients that represent the handlers for the devices on the robots such as sonar, SICK laser range-finders and color cameras" confirming the presence of Boyer's scholarship of integration. The simulate-validate approach has been used with great success in other projects such as "localization-space trails (LOST)" [31], a method that enables a team of robots to navigate between places of interest in an initially unknown environment using a trail of landmarks confirming the presence of Boyer's scholarship of application.

Research Paper 6

For our sixth research paper we can discuss [32] in which Player is also used in an investigation of multi-robot task allocation algorithms. Simply put, a method of dynamic task allocation for groups of such robots is presented providing a unified interface to a group of heterogeneous robots. In [32] an interesting question is posed by the authors one of whom (Brian P. Gerkey) is the developer of Player/Stage. The authors try to address the question of "How can we intelligently coordinate groups of robots?" arguing that the key to employ the potential of multi-robot systems is cooperation. Furthermore, they present a dynamic task allocation system called MURDOCH which uses Player in an investigation of multi-robot task allocation algorithms. The employment of Player throughout this research and its usage as an integral part of MURDOCH acknowledges the existence of Boyer's scholarship of integration. The technical details of this paper are beyond our discussion; it is our goal to show how Player/Stage nurtures research and in doing so show the evolution of ideas such as MURDOCH which have been a source of reference for further contribution in the field of robotics.

Research Paper 7

In [33] the topic of autonomous mobile agent navigation is discussed. This type of navigation is crucial to many mission-critical applications such as search and rescue. In simple terms this paper discusses how sensor networks may assist probabilistic roadmap methods (PRMs). There are direct acknowledgements in the paper that Player/Stage ideas were used throughout their project. For instance, the authors demonstrated that with their software they are in a position to simulate a "sensor network which can relay real-time temperature information from a spreading fire" [33]. The authors later on found that this facility along with a robot simulator [34] of Player/Stage, can effectively pilot a robot through fire when it is receiving real-time temperature data by its sensors confirming the existence of Boyer's scholarship of application. The authors claimed that by using a "border query strategy, [authors] can capture the spatiotemporal information at the reduced cost" [33]. This method and respective algorithms can effectively be employed in real scenarios as well which is the essence of the applicability of knowledge to

substantial problems in Boyer's model. It is also important to note that none of the four authors were part of the development team of Player/Stage, a testament to the fact that the outside research community has seen value in the base idea of Player/Stage in adopting it for further research contributions. This research paper [33]has been cited by 6 other articles according to Google Scholar which have been published in conferences such as IEEE International Conference on Robotics and Automation as well as the International Conference on Distributed Computing.

Research Paper 8

In [35] a new method for controlling a group of nonholonomic mobile robots to achieve predetermined formations without using global knowledge is presented. A sign of using Player/Stage idea as a base is evident in the simulation section of [35] where the usage of Player/Stage is explicitly mentioned to carry out a number of tests to display the "stability, robustness and agility of the proposed algorithm" confirming the existent Boyer's scholarship of integration. It is important to note that none of the four authors were part of the development team of Player/Stage. Moreover since the date of its publishing, this article has been cited by four other research papers published in IEEE in the field of Robotics.

Research Paper 9

Lastly we can track research papers such as [32] that we have discussed above specifically showing how further research has been built on papers that either used part of Player/Stage code or simply used its ideas. For instance, [32] was published in 2002 and as of June 6, 2008 according to Google Scholar it has been cited by 248 articles. When we take a closer look at these articles by going through their abstracts and list of their authors, we notice that there has been a continuous trend of articles published all the way from 2002 to 2008. These articles reference MURDOCH, not only by authors who are part of the development team of Player/Stage but also by new researchers who have seen value on the original idea of MURDOCH for further research. Through this process it is shown how influential the original idea of Player/Stage has been, creating an interesting

topic of discussion as it relates to the presence of one or more of Boyer's categories of scholarship.

For instance in 2004, [36] was published targeting research challenges in wireless sensor and actor networks. A quick look at the authors of this article confirms that they are not in the list of developers of Player/Stage showing that they have found value for further research based on MURDOCH's original idea which addressed the problem of multi-robot coordination. The authors have tried to answer the following question "Which actor(s) should execute which action(s)?" and found their answer in the "coordination" idea of [32] confirming the presence of Boyer's scholarship of integration as it relates to MURDOCH and Boyer's scholarship of application as it relates to the original idea of Player/Stage used by MURDOCH.

Concluding Remarks on Player/Stage

In this chapter, we have discussed why Player/Stage is one of the most widely used robot control interfaces in the world. We touched on the development of Player/Stage, showing how a research community has formed around it, and discussing nine research papers that have referenced the original idea.

The scholarship of teaching in Boyer's model was briefly touched on, relating it to the decision of the original authors to make the Player/Stage platform an Open Source model. Furthermore, in order to explore our claim for the candidacy of Player/Stage as an "infrastructure for a computing research community" and its relationship with Boyer's scholarship model nine research papers were considered. In each paper, the presence of one or more of Boyer's scholarship categories was discussed, making a case for both the existence of scholarship within Boyer's framework and for choosing Player/Stage as an "infrastructure for a computing research community".

CHAPTER 6: THE INSIGHT TOOLKIT (ITK)

The Insight Toolkit (ITK) is an Open Source software toolkit that is used for performing registration and segmentation. Segmentation can be described as the process of identifying and categorizing data obtained in a digitally sampled representation. Usually the sampled representation is an image obtained from such medical devices as CT or MRI scanners. Registration can be explained as the task of aligning or developing relationships between data. A practical example of the application of segmentation and registration in the medical field is to align a CT scan with a MRI scan for the purpose of combining the information contained in each scan.

In this section, the interesting history behind ITK is discussed along with a general overview of ITK's Open Source toolkit including its platform, projects building on the idea of ITK, research papers referencing the idea of ITK and commercial applications of ITK. It is the goal of this section to show that the Open Source model employed by ITK has had a direct effect in nurturing further research in the field of Medical Imaging. Lastly, a series of research papers along with concluding remarks are discussed portraying the existence of contribution (tools for a research community outside computing) and the connection to Boyer's model of scholarship.

History of ITK

In 1999, three commercial partners (GE Corporate R&D, Kitware Inc., MathSoft) and three academic partners (University of North Carolina (UNC), University of Tennessee (UT) and University of Pennsylvania (UPenn)) were granted three-year contracts to develop an Open Source registration and segmentation toolkit by the US National Library of Medicine of the National Institutes of Health (NIH) with the following goals (www.itk.org):

- Support the "Visible Human Project"

- Establish a foundation for future research
- Create a repository of fundamental algorithms
- Develop a platform for advanced product development
- Support commercial application of the technology
- Create conventions for future work
- Grow a self-sustaining community of software users and developers

A quick look at the nature of the goals set by NIH (www.ITK.org) can help us to draw comparisons with Boyer's model of scholarship. Specifically the first five objectives stated above are related more to the integration and application of knowledge and the last two relate to scholarship of teaching.

It is important to note that the Open Source registration and segmentation toolkit ultimately became the Insight Toolkit (ITK) and served as the foundation of the Insight Software Association. ITK's first official public release came out in 2002 after three yeas of collaboration between involved parties. Furthermore, the National Library of Medicine gave thirteen contracts to several entities in order to extend ITK's capabilities. This was indeed a sign that this platform was emerging and keeping its original anticipated momentum toward establishing itself as an Open Source software toolkit in the Medical Imaging field. The National Library of Medicine's funding continued all the way to 2008 with the latest version (3.6.0) released in April 14, 2008.

Technical Information

ITK is cross-platform (Unix, Windows and MacOSX) and is implemented in C++. It uses the CMake build environment to organize the compilation process as well as an automated wrapping process to create interfaces between C++ and interpreted programming languages such as Tcl, Java, and Python (using CableSwig). This structure allows developers to create software utilizing a range of programming languages.

Moreover, the design philosophy behind ITK provides data representation and algorithms for performing Segmentation and Registration. The primary focus as stated

previously is on medical applications, despite the fact that the toolkit is capable of handling other data types. Specifically the toolkit provides data representations for images (arbitrary dimension) and (unstructured) meshes while visualization and graphical user interface capabilities are supplied by other toolkits such as VTK , VisPack and 3DViewnix to name a few.

The Open Source Platform of ITK has enabled developers from around the world to debug, maintain, and extend the software. ITK uses extreme programming as a model for its software development with the specific goals of efficient communication and testing between developers in order to manage the rapid evolution of the software. ITK owes its stability to rigorous testing (using Dart) which measures the quality on daily basis, moreover the ITK Testing Dashboard is published frequently portraying its quality.

User Community

The Insight Community and Support ITK were developed from the beginning with the goal of creating a cohesive community effort in order to nurture Research (correlated with Boyer's scholarship of discovery), Teaching (correlated with Boyer's scholarship of teaching) and Commercial uses of toolkit (correlated with Boyer's scholarship of application). For users interested in ITK there are a number of possibilities to join as they can actively report bugs, defects in the system API or simply submit feature requests; this is currently being done through user's mailing lists. At this time, two mailing lists exist: the developers' and users' mailing list. The insight-users list is open to the public with the purpose of posting general questions, bug reports, or suggestions to improve ITK, and the insight-developers list is for developers with the purpose of developing and contributing customized classes.

Developers can also contribute classes or simply improve the existing base of classes in the toolkit. New developers can request permission to join the ITK developer mailing list and the pre-requisite of this privilege is demonstrating high levels of proficiency and honesty. In the course of research for ITK, we found a list of 29 developers who were active in the early stages of ITK development, however, as the code grows and becomes publicly accessible the most accurate way of determining the number

of developers in ITK is to view the CVS source code repository logs. Furthermore, research partnership with members of the Insight Software Consortium is indeed encouraged as funding can be obtained for high potential research ideas. This facility is further extended for educators as they may wish to use ITK in courses. Currently course materials are being prepared specifically for this purpose, which is indeed a sign of growing interest in this Open Source platform.

Related Projects

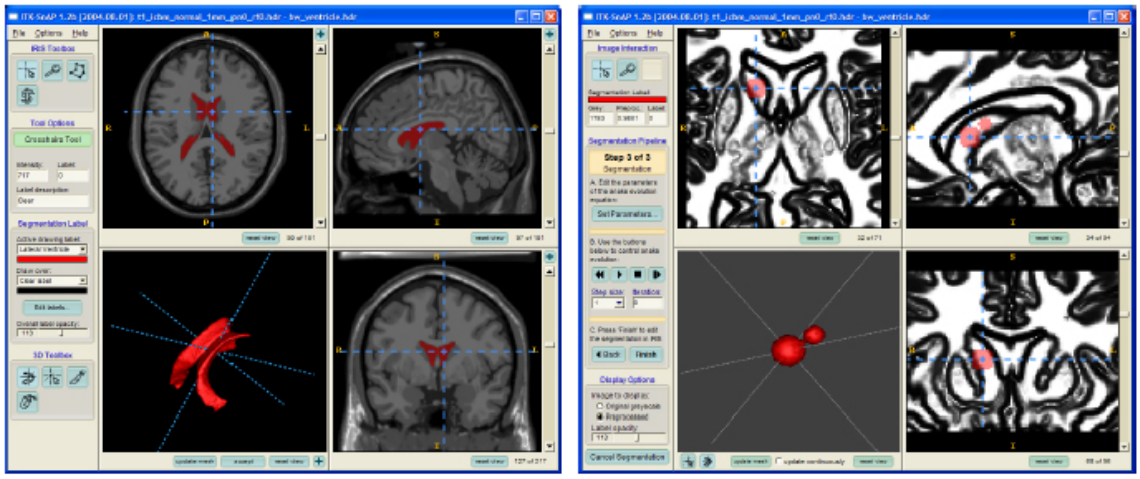
There are currently 21 applications listed on ITK's website (www.ITK.org) that have used the ITK toolkit as a base for their development. This would be a clear sign of the contribution of this Open Source platform which has caused the research community to spend time and effort in developing applications such as ITK-SNAP based on the original ITK platform. Simply put, development of such applications would have not been possible if ITK was not developed using Open Source model. Furthermore, development of such applications can be correlated with Boyer's scholarship of integration and scholarship of application as it is discussed in our chosen five research papers discussing ITK's platform.

ITK-SNAP

There have been numerous studies on active contour segmentation and its robust implementation using level sets, however the clinical research still heavily relies on manual slice-by-slice outlining for anatomical segmentation. To bridge the gap between methodological advances and clinical routine, ITK-SNAP has been developed to screen medical images with the purpose of defining and mining anatomical structures. This tool can be employed in two distinct modes of manual segmentation and semi-automatic segmentation. Segmentation with hand contouring is achieved via manual mode while semi-automatic mode is used to segment anatomical structures in three dimensions. It should be noted that this algorithm demands some level of guidance from the user with the easy to use interface that is provided by ITK-SNAP.

ITK-SNAP presents an easy to use image viewing user interface on top of which an automatic segmentation workflow is created. The image viewing UI (Figure 6.1) underlies the importance of the 3D nature of images by portraying three orthogonal views that cross each other at the position set by the 3D cursor [37]. This facility created by ITK-SNAP can be linked to Boyer's scholarship of integration based on algorithms developed in ITK's toolkit.

Figure 6.1: ITK-SNAP user interface



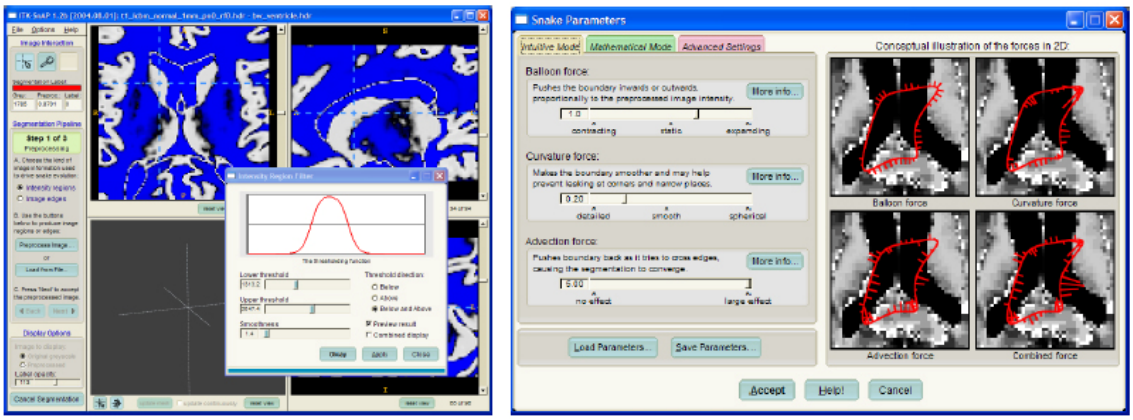
Left, ITK-SNAP user interface shows three orthogonal views of a volumetric image, linked by a common cursor. Right, the automatic segmentation workflow is organized as a wizard, shown here in the initialization stage. © 2005, Paul Yushkevich, by permission.

ITK-SNAP in Clinical Applications

The ITK-SNAP tool along with its trademark facilities namely user-guided 3D active contour segmentation, manual post-editing in 2D orthogonal slices, and 3D cutting knife capability is, more than ever, substituting for 2D slice editing in clinical image analysis. The ITK-SNAP tool is employed at Chapel Hill, Duke University and University of Pennsylvania for neuron imaging studies. The segmentation process is undertaken through two methods, either by using the fuzzy threshold option (shown in

Figure 6.2) which is used in the segmentation of the intracranial volume in head MRI or simply by employing tissue probability maps which are used for the segmentation of caudate nucleus structures employing grey matter probability map. Integration of ITK-SNAP in such procedures can be linked to Boyer's scholarship of integration.

Figure 6.2: Live feedback in ITK-SNAP



Left, a fuzzy two-sided threshold is being used to define object and background probability maps; the maps are displayed to the user of the fly. Right, parameter selection dialog for contour evolution. The effect of the parameters on the forces acting on the contour is illustrated in a 2D example. © 2005, Paul Yushkevich, by permission.

Also in other instances such as segmentation for liver, kidneys, bony structures and tumor, ITK-SNAP has proved to be useful in computer tomography imaging (CT). For instance, it has been employed to segment lung volumes in CT images and pulmonary vasculature in lung MR images proving not only the existence of Boyer's scholarship of integration based on original ITK libraries but also Boyer's scholarship of application in cases such as segmenting lung volumes. In terms of joint use in other applications we can point to deformable registration in several applications "including building a cortical atlas for template deformation morphology [38], automatic mask generation for 3D reconstruction of a murine brain from histology slices [39], and manual landmark definition." Such cases of direct integration of ITK-SNAP, yet again show the presence of Boyer's scholarship of integration based on ITK's platform. Other

commercial uses of ITK-SNAP, which are a sign of its practicality in the medical imaging field, include “manual post processing for automatic brain extraction tools and 3D smoothing of slice-by-slice manual segmentations with large step-edge artifacts” [37]. Such evidence of the usage of ITK-SNAP can be correlated with the fact that Boyer’s scholarship of application is present.

As we have mentioned, the ITK-SNAP application has been developed employing the ITK toolkit Open Source library for the purpose of screening medical images. In a way, we can look to ITK-SNAP as a secondary effort on top of an Open Source platform which has helped further research for publications such as [38] and [39]. We can call this effort a third level of further research in the field of medical imaging which has been directly caused by an Open Source platform namely the ITK toolkit. It seems as though Boyer’s scholarship of integration, in this case integrating ITK-SNAP, sets the stage for discovery of knowledge in research topics which directly discuss the results of such papers. Specifically [38] has been cited by seven other articles according to Google Scholar. Not surprisingly the name of Yushkevich as a lead researcher can been seen in three of the publications but the variety of topics provides a perspective that ITK-SNAP indeed has caused further research. For instance, [40] discusses that “corpus callosum [as] the largest white matter structure in the brain [is] of great interest in studies of brain connectivity”. The shape and area of the midsagittal section of the corpus callosum (MSCC) was “manually segmented from the FA images using ITK-SNAP”. Furthermore, it is explicitly mentioned that "semiautomatic and automatic techniques" of [38] were employed to divide each hemisphere of the cerebral cortex into four regions confirming the fact that a chain research effort has been caused by integrating ITK-SNAP.

Research articles referencing ITK

In our research, we have chosen five papers published in IEEE that have referenced the ITK toolkit extensively throughout their research. We will try to analyze how these papers have employed ITK toolkit ideas as well as discussing whether other publications have referred to their work or not, contributing to further research.

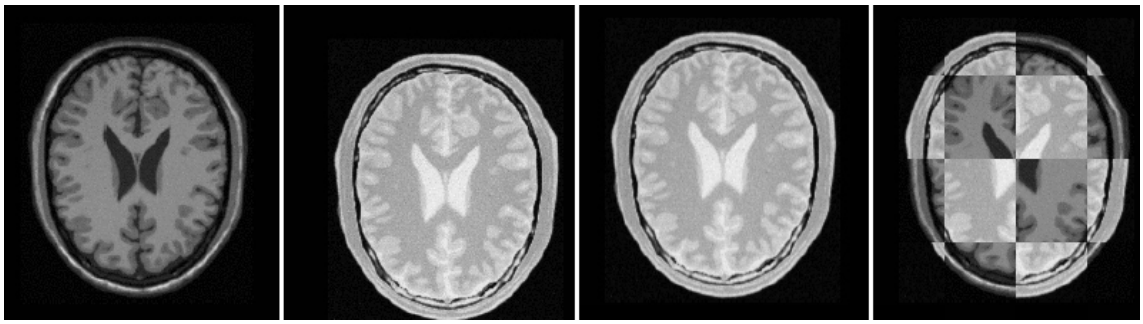
Furthermore, we will track the authors active in each paper to see how a research community has evolved surrounding each topic.

Research paper 1

In [41] the theory of mutual information and its application in the medical image registration is discussed. The authors argue that the registration of multimodality images is an integral part of medical image processing by proposing methods that can perform Registration of 2D images using Mutual Information. The authors explicitly mention that "registration uses methods from the Insight Software Toolkit" acknowledging the integration of knowledge from ITK's base code.

In order to show the validity of mutual information theory, two experiments based on ITK were conducted and they both showed high performance in the registration of 2D image using mutual information. In the first experiment, two human-brain images were employed as follows: a MRI image as the fixed image and a CT image as the moving image which was translated 13 pixels along the x axis and 17 pixels along Y axis.

Figure 6.3: MRI - Mapped moving image



The fixed image (left 1) is a MRI image, and the moving image (left 2), The mapped moving image (right 2) and the composition of fixed and moving images after (right 1) registration. © 2008, Shu Zhan, by permission

Experiment number two was conducted with only a minor revision. Again two human brain images were employed, a MRI image as the fixed image, and a CT image as

the moving image, which was translated 13 pixels along x axis, 17 pixels along y axis, with the difference that this time it was rotated 10 degrees clockwise (Figure 6.3). Due to the successful results of their experiments the authors point to the fact that their next project is using this method in the registration of 3D images as well as non-rigid images. This is a sign of maintaining existing momentum based on successful results which can be attributed to Open Source platform provided by ITK causing further research along the way, not only by way of integration of existing knowledge but also setting the stage for discovering new knowledge. Since this paper was published in 2008, I was not able to find publications referencing registration of 2D image using Mutual information. Nevertheless, the timeline shows that ITK to date is still nourishing further research based on its Open Source library for such methods as Mutual information used in this paper.

Research paper 2

The next paper [42], which was published in IEEE in 2006, discusses an Open Source software architecture for immersive medical imaging applications. For the purposes of our discussion in this thesis, it specifically employs Open Source and cross-platform libraries for medical applications of ITK with 3D graphic and virtual reality libraries such as Chromium to efficiently address the needs of both operators and researchers in the medical field proving the existence of Boyer's scholarship of integration based on ITK's toolkit. Moreover the authors of this system intelligently have enhanced the functionalities by "pervasive computing characteristics like context-awareness" and also providing transparent access with the goal of enabling mobile users to communicate with the virtual environment with their mobile devices.

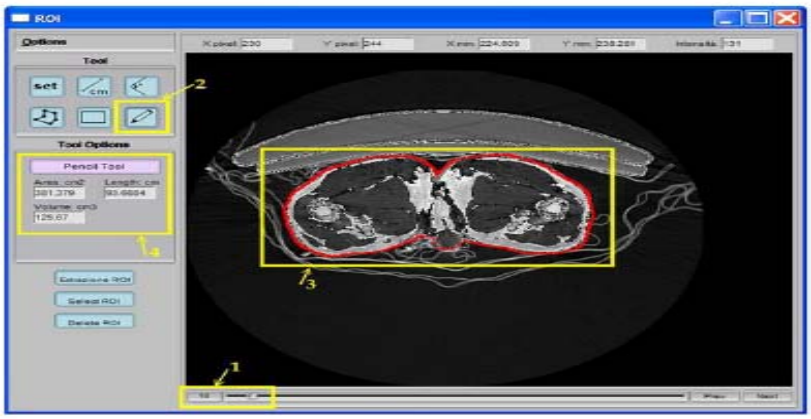
This paper [42] purposefully targets the current software limitations in the Medical Imaging field addressing the fact that many commercial tools are expensive and mostly designed for a specific hardware structure. The authors make a case that freeware applications that are mostly designed for a specific application field are not well documented thereby making it a difficult task to expand their existing proprietary libraries. Moreover, the authors argue that there is a need for new software architecture to

address a higher degree of interactivity within 3D space making a case for the objective of their research. This paper essentially proposes a new software architecture designed to address limitations mentioned above by creating its base on Open Source libraries to allow future integration of new software components and new devices such as data gloves and HMDs. Such initiative and recommendations can be discussed in the context of scholarship of integration. Integration of knowledge has provided the means for the authors to make their case for a new software with a “higher degree of interactivity”. Furthermore by pointing to limitations of existing software, the authors make a case for further research on devices such as data gloves, which could lead to discovery of new knowledge.

On a technical note, software architecture consists of Medical Imaging Service which provides users with functionalities for 2D and 3D visualization for image processing of DICOM files obtained from MRI, CT or PET instrumentation. A 3D Viewer provides a three-dimensional view of the model based on data obtained from 3D Model Reconstruction component and the Virtual Reality Device Handler (which is essentially a device management system) and has capabilities for supplying an interface to most virtual reality devices. If we zoom into the Image Processing component of this architecture we soon realize that this component integrates well-known implementations such as ITK and VTK libraries pointing to the presence of Boyer’s scholarship of integration.

Moreover, other than facilitating functionalities such as Region of Interest Selection, this architecture has created advanced ROI functionalities (Figure 6.4) “by which it is possible to apply a multi-selection function within a single image or to define a ROI of arbitrary shape by means of a set of user-defined control points”. This added enhancement would mean that the user could separate dimension, form, volume and position of the image in a very interactive fashion proving yet again the existence of Boyer’s scholarship of integration. One could argue further that since this architecture has created advanced ROI functionalities, the existence of Boyer’s scholarship of discovery is evident.

Figure 6.4: ROI Selection Panel.



Another interesting feature, which has been built on top of the ITK platform, is the image fusion. This enhancement enables the user to start from different modality images (i.e. PET and TAC) of the same object and then combine them into a single composite image for more accurate analysis and diagnosis supporting our case further for the fact that Boyer's scholarship of integration is present.

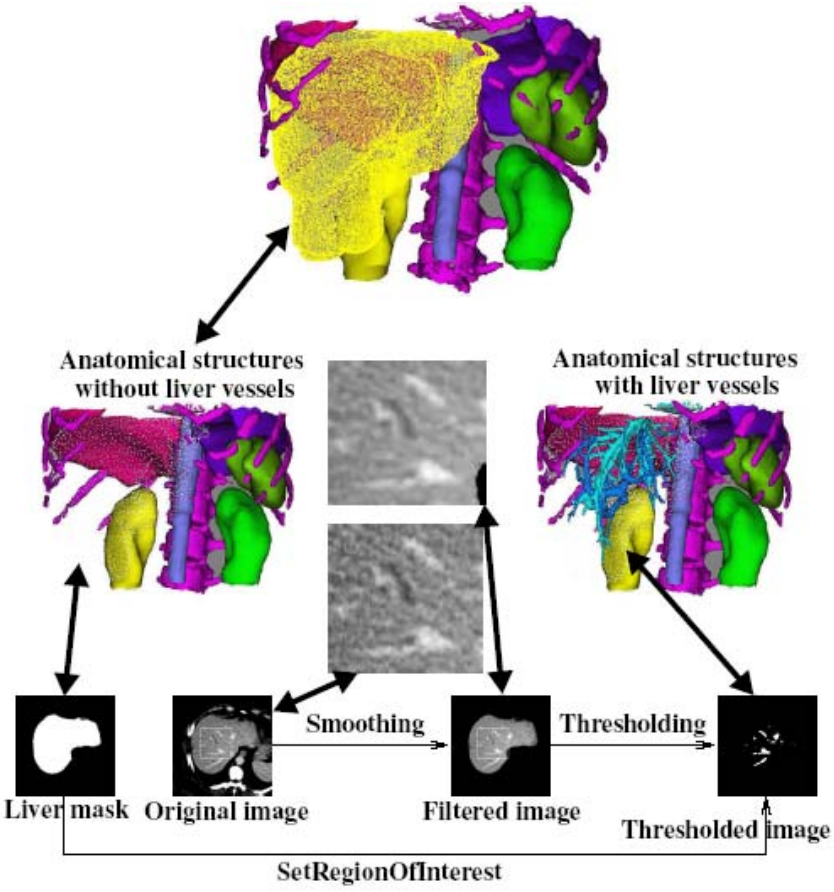
As far as articles referencing [42], we have found three articles which make references to the result of this publication. The timing of these articles are all recent, two have been published in 2008 and one in 2007 keeping in mind that two of the authors, namely Gallo and Marra, have appeared on all three references showing a sign of interest in topic for these specific researchers. Specifically, [43] proposes a new method for "the visualization and interaction with volumetric datasets in a fully immersive environment" employing the functionalities in VTK library and building on their previous work in [42], again pointing to the presence of Boyer's scholarship of integration as it relates to ITK's toolkit.

Research Paper 3

In [44] the enhancements made to ITK medical image processing library are discussed. The proposed enhancements extend the current ITK iterators leading to the creation of ITK++ framework, making a case for the presence of both Boyer's scholarship of integration and scholarship of discovery. Integration in a sense that ITK's

framework is enhanced and discovery in a context of creating a new framework, namely ITK++. The authors have evaluated their method by using it in a liver vessel segmentation from CT-scan images where focusing on liver area is a must in analyzing the overall image and thereby determining the accuracy of methods being employed. Through experimental results the authors show the enhancements in their algorithms namely discussing the fact that anisotropic filtering of the liver through their method was performed in only 16 seconds while the same process through ITK native framework would take more than 52 seconds. Furthermore, the authors discuss how the new code base remains manageable and efficient with a major advantage that ITK native library was not changed since the improvements were done in the form of add-ons.

Figure 6.5: Example of processing flow with proposed ITK++ design.



Authors have used simple image smoothing for segmentation along with intensity thresholding. Since the test case used was focused on internal liver structures (Figure 6.5), authors make the point that the procedure should be "applied to voxels belonging to the irregularly shaped area related to the liver, thus avoiding the unnecessary processing of external voxels". With the framework proposed in ITK++ this task is achievable and a visual demonstration of this process is portrayed in the figure above showing a significant performance improvement over the native ITK source code, once again proving the presence of integration of ITK's toolkit (original knowledge). Finally, we should mention that we could not find any research paper referencing the work of [44] using Google Scholar.

Research Paper 4

In [45] segmenting surgical needles as they are observed in x-ray fluoroscopy is discussed. The needle segmentation that is analyzed in this publication is essentially based on the image processing filters in the Insight Software Toolkit showing the presence of integration of knowledge (Boyer's scholarship of integration). This method proposes a "six degree-of-freedom robotic needle driver", exclusively for the purpose of medical applications. Through discussing their results the authors demonstrate how they have been able to "to detect and track needles in the laboratory environment and on x-ray images obtained from typical spine cases" employing segmentation methods of the Insight Software Toolkit.

One research paper that references [45] is [46] proposing a method to "[compute] optimal turning points to maximize the probability that the needle will reach the desired target" arguing this method is needed due to differences in patients as well as challenges in foreseeing needle-tissue interaction. Such recommendation is a direct cause for discovery of new knowledge in the context of Boyer's scholarship model in [46]. Authors of [46] furthermore employ the reasoning in [45] to argue that X-ray fluoroscopy is well suited for their application because "[X-ray fluoroscopy] generates 2D projection images from which the needle can be cleanly segmented", confirming the presence of integration

of knowledge. It is important to note that none of the authors of [46] were part of the original team that worked on [45], an indication that further research kept its momentum for a whole new team of researchers three years after the publication of [45].

Research Paper 5

In [47] an enhancement of a retinal blood vessel segmentation method employing the ITK toolkit is discussed. This enhancement is significant due to the fact that the “the appearance of the retinal blood vessels can be an important diagnostic indicator of various clinical systemic disorders of the eye and the body”. The Authors [47] argue that despite the fact that previous Matlab implementation for this segmentation was robust, nevertheless it couldn’t process high resolution images in recent databases. Authors make a case that their ITK version is capable of segmenting images between “600 and 1500 pixels squared size, around 8 times faster than the Matlab version” as well as handling higher resolution images. Furthermore using a public database, the authors demonstrate that not only does their ITK version segment 12 times faster than the Matlab version but it also has greater accuracy, confirming the presence of Boyer's scholarship of discovery.

Concluding Remarks on ITK

In this section we have analyzed one major application, namely ITK-SNAP that was created based on the ITK toolkit libraries. We have shown in detail how ITK has nurtured further research in the field of medical imaging as well as detailing out five research papers each addressing different medical imaging topics using ITK as a base of their research. We have made a case for the presence of Boyer’s scholarship of integration and scholarship of discovery as well as identifying the nature of the contribution (tools for a research community outside computing). Furthermore, ITK without a doubt has created a platform for researchers in both the medical field as well as computing science to build on proven result and practical applications that are used to provide solutions to real world problems, demonstrating the presence of Boyer’s scholarship of application in [45] and [46].

ITK's Open Source platform has created this means and facility for this growth and when we consider that we have only discussed one out of 21 applications of ITK, we can truly appreciate the scope of work that this platform covers. It seems that ITK has bridged the gap between several academic disciplines as it has evolved from its early years of conception. The partnership between Medical Schools and Computing Departments in different Universities is a testament that researchers can see a potential in gaining recognition by contributing to applications of ITK or through completely new research based on the original libraries available in the ITK toolkit. It would be very hard not to recognize the role of the Open Source platform of the ITK toolkit in nurturing this continuous stream of research activities for a sensitive field such as Medical Imaging. Thereby one can conclude that the success of ITK would not have been possible had it not been because of its Open Source platform.

CHAPTER 7: POWER SYSTEM ANALYSIS TOOLBOX

Power System Analysis Toolbox (PSAT) is an “open source Matlab and GNU Octave-based software package for analysis and design of small to medium size electric power systems” [48]. In order to discuss the types of scholarship contributions PSAT has made to the academic community, it would be prudent that we first provide background information on GNU Octave, which PSAT is based on.

GNU Octave is a high-level language designed for numerical computations; it is Open Source and can be modified under the terms of the GNU General Public License. It also has a long track record and an active user community surrounding it. In the following sections research papers referencing or employing techniques offered by or based on GNU Octave libraries are analysed (i.e. PSAT) to show concentrated effort by the research community in embedding this software in their work as they see practical value in employing its functionalities pointing to the existence of Boyer’s scholarship of integration and ultimately leading to discovery of new knowledge. This analysis also helps us to show the existence of contribution (robust implementation of important, previously published algorithm) and their connection to Boyer’s model of scholarship. Lastly, it is demonstrated that the Open Source nature of this software has been the main reason that GNU Octave has flourished to the point that it is currently being employed in research and development of toolboxes such as PSAT twenty years after its inception.

Research articles referencing PSAT

As we have indicated in the previous section, PSAT is a Matlab / GNU Octave toolbox for the purpose of analysis and simulation in electric power systems. This toolbox is currently being employed in few Universities and undergraduate courses and amongst its many functionalities we can point to main features such as power flow estimation and small signal stability analysis [48]. In the following sections, five research papers that have referenced PSAT extensively are discussed touching on topics such as

"two-point estimate method (2PEM)" [49] and "Flexible Alternating Current Transmission System (FACTS)" [51] all employing PSAT functionalities.

Research Paper 1

The first paper for the purpose of our discussion is [48] which describes the Power System Analysis Toolbox (PSAT). PSAT is an "Open Source Matlab and GNU Octave-based software package for analysis and design of small to medium size electric power systems". Integration of PSAT in the analysis of such systems can be linked to the presence of Boyer's scholarship of integration, facilitated by GNU Octave libraries. Since most of our discussion from now on will be based on PSAT, it will be prudent to discuss a few research papers that have employed PSAT extensively in order to drive and simulate their result. [48] was published in 2005, and has only one author who is not part of the list of contributors in the development of GNU Octave. According to Google Scholar, [48] has been cited by 34 other research papers and it is the intent of this section to show how diverse the list of references is with respect to the field of research and the way PSAT has been employed in their work. Lastly, it should be mentioned that the Matlab environment is a commercial product, which means its kernel and libraries cannot be modified nor freely distributed. In light of this issue in order to create a platform for further research and potential contribution by programmers active in this arena, both the toolbox and the platform on which the toolbox runs should be free. In order to achieve such a task PSAT can run on GNU Octave, which is essentially a free Matlab clone, making the process of integration easier.

Research Paper 2

The second paper of interest that that has employed PSAT is [49], which presents an application of a "two-point estimate method (2PEM) to account for uncertainties in the optimal power flow (OPF) problem in the context of competitive electricity markets." The authors argue that the advantage of 2PEM is that it is not dependent on derivatives of the non-linear function and further argue that their obtained results are more accurate compared to MCS. After detailing out how different standard deviations were embedded

to portray market participant's behavior, the authors explicitly mention the use of "PSAT to solve required OPFs." Such a statement for a research paper which has been published in IEEE and has been referenced by 9 other articles all in the same field and published in IEEE shows an importance of PSAT as an instrumental component of this research. Lastly, it confirms the existence of Boyer's scholarship of integration as it relates to solving OPF by employing PSAT functionalities.

Research Paper 3

The third paper that has employed PSAT is [50], which proposes an emergency control system based on multi-agent technique. The presented system has the capability to find a coordinated control action, which might be load shedding or generation change, in order to "prevent the violation of power system voltage stability". The authors make explicit references to the use of PSAT in their research and make it clear that the test power system was indeed simulated in PSAT software confirming the existence of Boyer's scholarship of integration. Furthermore, the authors discuss the efficiency of their proposed technique by numerical simulation through PSAT under different power system conditions and variety of emergency situations. Driving such results without the use of PSAT would seem to be a daunting task as the authors carried out "power flow and optimization computations" using PSAT. Again it seems that PSAT has been an integral part of this research as the authors make no hesitation to mention that their targeted "LIB point was determined by the Continuation Power Flow (CPF) method implemented in PSAT Toolbox". One can immediately determine that PSAT is easily an integral component of this research as certain parameter and results simply would not have been possible to determine with comparable software available at the disposal of the authors. Such acknowledgements further support the existence of Boyer's scholarship of integration and PSAT's contribution along the way as a "robust implementation of important, previously published algorithm". It is interesting to note that none of the authors were part of the official contributor list in GNU Octave code base. It seems that GNU Octave has become not only a platform to which researchers can add by participating through submitting modules, but also it has become an instrumental piece of

the puzzle in most of the academic research of this kind that need complex computations in order to drive its results and draw its conclusions.

Research Paper 4

The fourth paper of interest as it relates to our discussion of PSAT is [51] which focuses on the efficient usage of a Flexible Alternating Current Transmission System (FACTS) device called Unified Power Flow Controller (UPFC) for power flow control. This research paper demonstrates through a case study how PSAT has been employed for "network analysis of alternative means of improving existing transmission capability" by driving its solutions and desired results using PSAT. The applicability of PSAT as a computational tool is discussed at length in this paper analyzing effective simulation and monitoring the FACT device. Furthermore it is explicitly mentioned that the development of the network model is based on "using the m-files with PSAT coding" which leads to the point, once again, that PSAT is a cornerstone of this research as both simulation and sensitive results are dependent on its usage confirming the presence of Boyer's scholarship of integration. This claim is supported by more evidence throughout the paper as power flow solution is derived on the main PSAT GUI screen, not to mention that the IEEE 14-bus network voltage simulation is performed using PSAT as well. Lastly, authors mention that their application namely UPFC in PSAT "has given an immediate remedy for solving high voltage transmission in long transmission line" which is achieved by controlling UPFC variables through PSAT GUI. Once again, if we take a look at the authors' names, we notice that none of them are part of the list of contributors to original GNU Octave but yet they have relied heavily on an application which is based on GNU Octave. This simple observation shows the power of an application which has been created through the contribution of many researchers over the course of 20 years and it has evidently become a vital part of [51] in a specialized field of study.

Research paper 5

The fifth paper of interest which utilizes PSAT for simulation purposes is [52] which discusses recent advancements made in the efficiency of hybrid electric vehicles

(SHEVs). Such improvement in efficiency has been achieved by focusing the operation of the engine only on optimal efficiency regions employing sliding mode controllers (SMCs). The proposed method in this research paper is tested on a computer simulation model of the XM1124 prototype using PSAT. The authors have used PSAT as a comparison method between actual and simulated data during the field-testing of the prototype and the results have shown that "PSAT simulation model could be used with minor modifications to provide a baseline upon which alternate components, configurations, and control strategies can be built and evaluated". Such an acknowledgement for the usage of PSAT points to the presence of Boyer's scholarship of integration. Once again, we notice that none of the authors of this IEEE publication are part of the team that has contributed to GNU Octave but yet the use of PSAT as an integral component of their research is evident. Upon closer inspection we have noticed that despite the fact that this article was published in 2006, 9 major articles published in IEEE have referenced the work of the authors leading to further research in their respective field. Such momentum would have not been possible without the use of PSAT and in turn the contribution of researchers to the GNU Octave free code base.

Concluding remarks on PSAT

In this section, an overview of GNU Octave - an Open Source platform with a twenty-year track record that has gone through many revisions was provided. GNU Octave has proved its applicability as it has been employed in academic research as well as commercial applications. Toolkits such as PSAT which have been developed based on GNU Octave, have created a whole new facility for researchers as we have seen in the papers discussed above portraying the presence of Boyer's scholarship of integration. The topic of academic research employing this Open Source library is so diverse that one can easily be convinced of its contribution along the way. Such success would not have been possible without the Open Source platform structure of GNU Octave in which multiple contributors have added functionalities in order to equip the academic community to not only prove but also simulate their results using PSAT.

CHAPTER 8: CONCLUDING REMARKS

In this thesis, we have tried to raise the importance of scholarship in Open Source software. This report has presented Boyer's expanded model of scholarship and observed that the qualities of scholarship in Boyer's model are present in Open Source software projects. Analysis of thirty-two Open Source software projects suggested that Open Source software projects can be classified based on three types of contributions: infrastructure for a computing research community, tools for a research community outside computing and robust implementation of important, previously published algorithms. Three representative case studies, Player/Stage (robotics), the Image Processing Toolkit (image processing) and PSAT (numerical computation) have been discussed in detail to illustrate the presence of attributes of scholarship and the three types of contribution.

Through our introductory sections, we have created a platform for the discussion of scholarly contribution in Open Source software. We have also discussed Boyer's expanded definition of scholarship and explored how the current focus on funded research fits within this model. Moreover, we have identified the types of contributions that are found in Open Source software and made a case for why it is important to consider scholarship in Open Source software. The degree of contribution (in Open Source software) was not addressed, but instead the existence of contribution within one or more Boyer categories was shown. In this context, the three case studies were analysed in detail.

In the section discussing Player/Stage, after touching on the development of Player-Stage we have shown how a research community has formed around the original idea of Player/Stage. We have supported our claim by going through nine research papers that have referenced the original idea, showing the presence of Boyer's scholarship of integration as well as Boyer's scholarship of application. Furthermore, in order to explore our claim for the candidacy of Player/Stage as an "infrastructure for a computing research

community” and its relationship with Boyer’s scholarship model, we have shown the presence of one or more of Boyer’s scholarship categories in each of the proposed research papers. Through this process we have made a case for both the existence of scholarship within Boyer’s framework and the choice of Player/Stage as an “infrastructure for a computing research community”.

In the section discussing ITK, it is shown that the Open Source model employed by ITK has had a direct effect in nurturing further research in the field of Medical Imaging. A series of research papers along with concluding remarks were discussed portraying the type of contribution (tools for a research community outside computing) and the connection to Boyer model of scholarship. We have analyzed one major application, namely ITK-SNAP that was created based on ITK toolkit libraries. We have shown in detail how ITK has nurtured further research in the field of medical imaging as well as detailing five research papers each addressing different medical imaging topics using ITK as a base of their research. We have made a case for the presence of Boyer’s scholarship of integration and scholarship of discovery as well as portraying the type of contribution (tools for a research community outside computing). Furthermore, we have shown how ITK has created a platform for researchers in both the medical field as well as computing science to build on proven results and practical applications that are used to provide solutions to real world problems demonstrating the presence of Boyer’s scholarship of application.

In the section discussing PSAT, we touched on GNU Octave and discussed how PSAT has helped further research in the numerical computation field. Research papers referencing or employing techniques offered by PSAT were analysed to show concentrated effort by the research community in embedding this software in their work making a case for the type of Boyer’s scholarship of integration. This analysis also helped us to show the existence of contribution (robust implementation of important, previously published algorithm) and the connection to Boyer’s model of scholarship.

Lastly, it should be noted that despite a thorough research on SPLASH-2, a candidate for “robust implementation of important, previously published algorithm”, we decided to move this case study to the appendix section as we chose to discuss PSAT

instead. SPLASH-2 at best was a benchmark suite which by its nature did not really change over time and was employed mostly for the analysis of “centralized and distributed shared address space multiprocessors” [53], leaving us with not many research papers to analyse. On the other hand PSAT had multiple functionalities and was employed in a wide array of research related to “electric power systems” [48], helping us making a case for its candidacy in “Robust implementation of important, previously published algorithm” category.

APPENDICES

Appendix A: Repositories

1. **SourceForge.net** (http://sourceforge.net/): It is the world's largest Open Source software development web site. SourceForge.net provides free hosting to Open Source software development projects with a centralized resource for managing projects, issues, communications, and code.
2. **The Apache Software Foundation** (http://www.apache.org): It provides support for the Apache community of Open Source software projects. The Apache projects are characterized by a collaborative, consensus based development process, an open and pragmatic software license, and a desire to create high quality software that leads the way in its field.
3. **The Computational Infrastructure for Operations Research - COIN** (http://www.coin-or.org/): This project is an initiative to spur the development of Open Source software for the operations research community.
4. **IBM** Resources for Open Source development and implementation http://www.ibm.com/developerworks/opensource/
5. **Sun Micro System** Open Source: http://www.sun.com/software/opensource/index.jsp
6. The source for **Java** Technology collaboration: http://www.java.net/
7. **Freshmeat** (http://freshmeat.net/): Unix software directory
8. **CPAN** (http://www.cpan.org/): Perl module directory
9. **Jakarta** (http://jakarta.apache.org): The Jakarta Project offers a diverse set of open source Java solutions and is a part of The Apache Software Foundation

(ASF) which encourages a collaborative, consensus-based development process under an open software license.

10. **Savannah** (http://savannah.gnu.org): It is a central point for development, distribution and maintenance of GNU Software.
11. **CTAN** (http://www.ctan.org/): The Comprehensive TeX Archive Network is the authoritative collection of materials related to the TeX typesetting system.
12. **Sybase Open Source** (http://www.sybase.com/developer/opensource): Sybase has been a leader in developing and expanding innovative database technology.
13. **Novell Forge** (http://forge.novell.com/modules/news/) collaboration Web site for development of Open Source project
14. **Intel Open Source** (http://www.intel.com/cd/software/products/asmo-na/eng/219778.htm)
15. **Real Networks** (https://helixcommunity.org/): The Helix Community is an open collaborative effort among multimedia enthusiasts to develop and extend the Helix DNA™ platform. The community offers a collaborative, consensus based development process, with an extensive and diverse ecosystem of developers all involved in creating world class multimedia enabled applications. The Helix Community was created in July of 2002 by Real to help the industry standardize on an open digital media platform.
16. **HP Open Source** (http://opensource.hp.com/) HP has been an active sponsor of numerous Open Source and Linux organizations towards the stewardship and success of Open Source technology and its vast user community. Organizations such as the Free Software Foundation, Open Source Software Institute, and Linux Foundation are all supported by HP.
17. **Network Associates** (http://opensource.nailabs.com/)
18. **Thoughtworks** (http://opensource.thoughtworks.com/index.html): ThoughtWorks has a very high concentration of Open Source developers in its ranks. Here users can find a comprehensive resource of ThoughtWorkers' contribution to Open Source.

19. **Mitre** (http://www.mitre.org/work/tech_transfer): The MITRE Corporation is a not-for-profit organization chartered to work in the public interest. As a national resource, they apply their expertise in systems engineering, information technology, operational concepts, and enterprise modernization to address our sponsors' critical needs.
20. **DSpace** (http://www.dspace.org): DSpace captures user's data in any format – in text, video, audio, and data. It distributes it over the web. It indexes user's work, so that it can be searched and retrieved. It preserves digital work over the long term and it provides a way to manage user's research materials and publications in a professionally maintained repository to give them greater visibility and accessibility over time.
21. **Fedora** (http://fedoraproject.org/): It is a Linux-based operating system that showcases the latest in free and Open Source software.
22. **EPrints** (http://www.eprints.org/): It is the most flexible platform for building high quality, high value repositories, recognised as the easiest and fastest way to set up repositories of research literature, scientific data, student theses, project reports, multimedia artefacts, teaching materials, scholarly collections, digitised records, exhibitions and performances.
23. **CDSware** (http://cdsware.cern.ch): The CERN Document Server Software Consortium produces document and conference management software since 1995.
24. **Google Source Repositories** (http://code.google.com/hosting/) Google's hosting service, which accumulated dozens of new projects on its opening day, features mechanisms to store software, discuss it with mailing lists and track bugs. Google permits projects under a variety of Open Source licenses--but not the full range.
25. **Ohloh** (http://www.ohloh.net): Ohloh is an Open Source network that connects people through the software they create and use.

Appendix B: SPLASH-2

SPLASH-2 which is employed to facilitate the analysis of “centralized and distributed shared address space multiprocessors” [53], was part of the original thesis under the category of Open Source benchmark suite. It was used as an example to show how an active community of users employ this benchmark extensively. Because of a change in the structure of my thesis, this Open Source benchmark did not correlate with the other three case studies and their associated contribution. Due to the considerable amount of research done in this section, I saw fit to report my findings in the appendix section for the benefit of other researchers. In short, this benchmark suite is designed to develop parallel programs for the purpose of evaluation of architectural proposals and trade offs. SPLASH-2 was selected in the “robust implementation of important, previously published algorithm” contribution category. The authors of [53] point to the inherent difficulty of analysis of centralized and distributed shared-address due to the nature of variables that one has to deal with. Specifically, the authors mention that dealing with “system parameters such as cache size, associativity, and line size can both quantitatively and qualitatively impact the results of a study.”

Splash-2 Key research papers

The authors of [[53] point to two primary goals in their paper: first to quantitatively characterize SPLASH-2 programs as it relates to central properties and architectural communication of its modules such as computational load balance and set sizes. The second goal is of a methodological nature in which the authors try to assist people employing their model use it efficiently by describing how operating points and cache sizes should be defined to represent real practical situations.

The five authors of [53] published their paper in 1995 and since then, their paper has been cited by 353 other research papers according to ACM digital library as well as 1611 citations according to Google Scholar. This in itself can be a clear sign of contribution to the academic community nurturing further research in this field of study. In order to confirm this fact, I have randomly chosen twenty articles out of this list to determine whether SPLASH-2 has indeed played a central role in these research papers.

The result should provide us with an indication of the degree in which SPLASH-2 was relied upon in the cited articles.

The first article of interest is [54] which discusses a non-blocking FIFO queue algorithm for multiprocessor shared-memory systems. The authors indicate that on the analysis of their SGI multiprocessor, it “was possible to use the radiosity from SPLASH-2 shared-address-space parallel applications”. The same trend holds true in [55] in which SPLASH-2 was used in the evaluation of Filtering for the Intel Thread Checker Race Detector. Moving on to our third candidate which proposes A Chip-Multiprocessor with Transactional Memory Support (ATLAS), we can clearly see that in order to demonstrate the performance potential of ATLAS three scientific benchmarks (radix, mp3d and ocean) from SPLASH-2 were used. Candidate four [56] has employed SPLASH-2 in the same way by selecting six benchmarks from the SPLASH-2 suite for the purpose of evaluating their Boosting Superpage Utilization method. Candidate five [57] which discusses Cycle Accurate Transaction-driven Simulation (CATS) uses SPLASH-2 benchmarks "to evaluate the CATS framework".

The trend of employing SPLASH-2 holds true in [58] where Coherence Controller Architectures for SMP-Based CC-NUMA Multiprocessors are discussed in candidate six. Moving forward, in candidate seven [59] we can see SPLASH-2 involved to a lesser degree compared to previous research papers in the discussion of Java Multithreaded Applications as it was not the central benchmark used in conducting the research. Nevertheless, the results driven from this research by SPLASH-2 were compared to synthetic workload applications such as MTD. Candidate eight [60] discussing Optimized CC-NUMA system has frequently uses the SPLASH-2 suite as a benchmark to “evaluate the performance of the shared memory multiprocessor systems”. The trend of employing SPLASH-2 as a benchmark continues and we see the same pattern in candidate nine [61]. In discussion of [62] as our candidate number ten, test workloads Raytrace and ocean taken from the Splash-2 suite were also employed to analyze implementing efficient fault containment for multiprocessors.

In order to recap our progress so far, all of our ten randomly chosen candidates of the list of 353 citations have demonstrated that they have used SPLASH-2 suite

applications as benchmarks in their studies. In our further analysis, we noticed that candidate eleven [63] in our list has simply used the concepts of SPLASH-2 in its research paper as opposed to employing application in SPLASH-2 for benchmark purposes. The same holds true for candidate number twelve [64] discussing the integration of non-blocking synchronisation in parallel applications employing SPLASH-2 concepts. Moving forward to candidate thirteen [65] we are back to observing SPLASH-2 being used as a benchmark specifically mentioning that "[in their] multiprocessor experiments, [authors] use the SPLASH-2 parallel benchmark suite". Candidate fourteen [66] specifically mentions usage of SPLASH-2 in their discussion of Reducing Network Interface Memory Requirements. Moving to candidates fifteen [67] and sixteen [68] we see the explicit usage of SPLASH-2 as a benchmark application helping the authors to compare their results and draw their conclusions. Candidate seventeen [69] which discusses simultaneous partitioning didn't reference SPLASH-2 explicitly in its discussion of the benchmark but I noticed the "water" application being used as a benchmark which is one of the applications of SPLASH-2. Candidate eighteen [70] researching Symmetry and performance in consistency protocols employed all nine applications from SPLASH-2 suite. Candidates nineteen [71] and twenty [72] follow the trend and make clear indications that SPLASH-2 was used as their benchmark throughout their study. One quick look at the list of our twenty candidates shows that aside from two articles that used SPLASH-2 concepts the other eighteen candidates have made explicit references to the usage of the SPLASH-2 suite in conducting their research. This 90% plus success rate in our randomly selected articles would most probably hold true for the rest of the 333 cited articles in ACM digital library as SPLASH-2 has the capacity and depth to be used in many shared memory applications considering its four Kernel and nine applications that are repeatedly being referenced in academic research.

REFERENCE LIST

[1] Available: Wikimedia Foundation, Inc. http://en.wikipedia.org/wiki/Scholarship [Accessed: April 2, 2009].

[2] Achinstein P, Science Rules: A Historical Introduction to Scientific Methods. 1st ed.: Hopkins Fulfillment Service; 2004.

[3] Denning PJ, Designing new principles to sustain research in our universities. Commun ACM 1993;36(7):98-104.

[4] Denning PJ. A new social contract for research. Commun ACM 1997;40(2):132-134.

[5] Emanuel L, Rennie D., Yank V. When authorship fails. A proposal to make contributors accountable. 1997; 278(7).

[6] American Psychological Association. Publication manual of the American Psychological Association. 2001:350.

[7) Greene M, The demise of the lone author. Nature 2007 Dec 20; 450(7173):1165.

[8] Price D. J. S. Collaboration in an Invisible College. In Little science, big science...and beyond. 1986:119-134.

[9] Regalado A, Multiauthor papers on the rise. Science. 1995 May 12;268(5207):25.

[10] Committee on Science, Engineering and Public Policy, National Academy of Sciences. On Being A Scientist: Responsible Conduct In Research. 1995 .

[11] Aakvaag A, Andersen D., Dahiquist G., Nylenna M., Sarvas M. Handling of scientific dishonesty in the Nordic countries. Lancet, London, ROYAUME-UNI.1999 ;354:11-18.

[12] Boyer EL, Scholarship Reconsidered: priorities of professoriate. : The Carnegie Foundation for the Advancement of Teaching; 1997.

[13] Guthrie JW. Encyclopedia of Education. 2003.

[14] Perens B. Open Sources: Voices from the Open Source Revolution. O'Reilly 1st ed.; January 1999.

[15] Landry J. Profiting from Open Source. Harvard Business Publishing. 2000.

[16] Gupta R, Plotkin H, What (and Why) you should know about open-source software. Harvard Management Update. 1998;12:8-9.

[17] Payne C, On the Security of Open Source Software. Info Systems Journal 2002;12(1):61–78.

[18] Josh J, Josh L, Tirole, The Simple Economics of Open Source. Harvard Business School. 2000.

[19] Hann I, Roberts J, Slaughter S. Why Do Developers Contribute to Open Source Project? First Evidence of Economic Incentives. Proceedings of the 24th International Conference on Software Engineering. 2002.

[20] Angelis L, Bleris GL, Oikonomou A, Stamelos I, Code Quality Analysis in Open Source Software Development. Info Systems Journal 2002;12(1):43–60.

[21] Gallivan MJ, Striking a Balance Between Trust and Control in a Virtual Organization: A Content Analysis of Open Source Software Case Studies. Info Systems Journal 2001;11(4):277–304.

[22] Available: Wikimedia Foundation, Inc. http://playerstage.sourceforge.net/wiki/PlayerUsers. [Accessed: March 19, 2009].

[23] Available: Wikimedia Foundation, Inc. http://playerstage.sourceforge.net/wiki/PlayerHistory. [Accessed: March 9, 2008].

[24] Gerkey B. P, Howard A, Vaughan R. T, The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In Proceedings of the 11th International Conference on Advanced Robotics; 2003.

[25] Glassick CE, Huber MT, Maeroff GI. Scholarship Assessed: Evaluation of the Professoriate. THE CARNEGIE FOUNDATION 1st ed.; 1997.

[26] Fritsch J, Haasch A, Hofemann N, Sagerer G. A multi-modal object attention system for a mobile robot. IEEE/RSJ International Conference on Intelligent Robots and Systems 2005:2712-2717.

[27] Christensen HI, Topp EA,. Tracking for following and passing persons. IEEE/RSJ International Conference on Intelligent Robots and Systems 2005;2(6):2321-2327.

[28] Jung B, Sukhatme G. Tracking targets using multiple robots: The effect of environment occlusion. Autonomous Robots 2002;13(3):191-205.

[29] Dahl TS, Mataric MJ, Sukhatme GS. Adaptive spatio-temporal organization in groups of robots. IEEE/RSJ International Conference on Intelligent Robots and System 2002;1:1044-1049.

[30] Mataric MJ, Sukhatme GS, Tews AD, A scalable approach to human-robot interaction. IEEE International Conference on Robotics and Automation 2003;2:1665-1670.

[31] Mataric MJ, Stoy K, Sukhatme GS, Vaughan RT, LOST: localization-space trails for robot teams. IEEE Transactions on Robotics and Automation 2002;18(5):796-812.

[32] Gerkey BP, Mataric MJ. Sold!: auction methods for multirobot coordination. IEEE Transactions on Robotics and Automation 2002;18(5):758-768.

[33] Alanku G, Atay N, Bayazit OB, Chenyang L,. Spatiotemporal query strategies for navigation in dynamic sensor network environments. IEEE/RSJ International Conference on Intelligent Robots and Systems 2005:3718-3725.

[34] Gerkey BP, Howard A, Mataric MJ, Stoy K, Sukhatme GS, Vaughan RT, Most valuable player: a robot device server for distributed control. IEEE/RSJ International Conference on Intelligent Robots and Systems 2001;3:1226-1231.

[35] Guangming Xie, Jinyan Shao, Junzhi Yu, Long Wang. A tracking controller for motion coordination of multiple mobile robots. IEEE/RSJ International Conference on Intelligent Robots and Systems 2005:783-788.

[36] Akyildiz I, Kasimoglu I. Wireless sensor and actor networks: Research challenges. Ad Hoc Networks Journal (Elsevier). 2004;2(4):351-367.

[37] Cody H, Gee JC, Gerig G, Ho S, Piven J, Yushkevich P, User-Guided Level Set Segmentation of Anatomical Structures with ITK-SNAP. MICCAI Open-Source Workshop 2005.

[38] Dubb A, Gee J, Gur R, Xie Z, Yushkevich P, Regional structural characterization of the brain of schizophrenia patients. Academic Radiology 2005;3217:688-695.

[39] Avants BB, Burstein PD, Hawrylycz M, Ng L, Yushkevich PA, Zhang H, et al. Using MRI to build a 3D reference atlas of the mouse brain from histology images. Proc. Intl. Soc. Magn. Res. Med. 2005:2809.

[40] Cook PA, Duda JT, Hui S, Simon TJ, Yushkevich PA, Zhang H, et al. Shape-Based Normalization of the Corpus Callosum for DTI Connectivity Analysis. Medical Imaging, IEEE Transactions on 2007;26(9):1166-1178.

[41] Jiang J, Li H, Shi Z, Zhan S. 2D Multimodality Image Registration Based on ITK by Mutual Information. The 2nd International Conference on Bioinformatics and Biomedical Engineering 2008:2365-2368.

[42] Coronato A, De Pietro G, Marra I. An Open-source Software Architecture for Immersive Medical Imaging. IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems 2006:166-170.

[43] De Pietro G, Gallo L, Marra I, Vanzanella C. A New Approach For Handling 3D Medical Data In An Immersive Environment. IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems 2007:63-66.

[44] Agnus V, Fasquel J-. Improving genericity and performances of medical systems based on image analysis. 18th IEEE Symposium on Computer-Based Medical Systems 2005:247-252.

[45] Cleary K, Corral G, Ibanez L, Navab N, Patriciu A, Stoianovici D. Segmentation of surgical needles for fluoroscopy servoing using the insight software toolkit (ITK). Engineering in Medicine and Biology Society, 2003. Proceedings of the 25th Annual International Conference of the IEEE 2003;1:698-701.

[46] Alterovitz R, Branicky M, Goldberg K. Constant-curvature motion planning under uncertainty with applications in image-guided medical needle steering. Workshop on the Algorithmic Foundations of Robotics 2006.

[47] Hughes AD, Parker KH, Perez MEM, Thorn SA. Improvement of a retinal blood vessel segmentation method using the Insight Segmentation and Registration Toolkit (ITK). Annual International Conference of the IEEE Engineering in Medicine and Biology Society 2007:892-895.

[48] Milano F. An Open Source Power System Analysis Toolbox. IEEE Transactions on Power Systems. 2005;20(3):1199-1206.

[49] Canizares CA, Verbic G. Probabilistic Optimal Power Flow in Electricity Markets Based on a Two-Point Estimate Method. IEEE Transactions on Power Systems 2006;21(4):1883-1893.

[50] Etingov PV, Panasetsky DA, Voropai NI. Multi-agent approach to emergency control of power system. Third International Electric Utility Deregulation and Restructuring and Power Technologies 2008:2157-2161.

[51] Al-Dabbagh M, Thum PC, Yap EM. Applications of FACTS Controller for Improving Power Transmission Capability. IEEE Region 10 TENCON. 2005:1-6.

[52] Bogosyan S, Goering DJ, Gokasan M. Sliding mode based powertrain control for efficiency improvement in series hybrid-electric vehicles. IEEE Transactions on Power Electronics 2006;21(3):779-790.

[53] Gupta A, Ohara M, Singh JP, Torrie E, Woo SC. The SPLASH-2 programs: characterization and methodological considerations. 22nd Annual International Symposium on Computer Architecture 1995:24-36.

[54] Tsigas P, Zhang Y. A simple, fast and scalable non-blocking concurrent FIFO queue for shared memory multiprocessor systems. Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures 2001:134-143.

[55] Bliss BE, Ma Z, Petersen P, Sack P, Torrellas J. Accurate and efficient filtering for the Intel thread checker race detector. Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures 2006:34-41.

[56] Chung J, Park CH, Park D, Roh Y, Seong BH,. Boosting superpage utilization with the shadow memory and the partial-subblock. Proceedings of the 14th international conference on Supercomputing 2000:187-195.

[57] Ha S, Kim D, Gupta R. CATS: cycle accurate transaction-driven simulation with multiple processor simulators. Design, Automation & Test in Europe Conference & Exhibition 2007:749-754.

[58] Lim B, Michael MM, Nanda AK, Scott ML. Coherence controller architectures for SMP-based CC-NUMA multiprocessors. Proceedings of the 24th annual international symposium on Computer architecture 1997:219-228.

[59] Choi J, Srinivasan H. Deterministic replay of Java multithreaded applications. SIGMETRICS symposium on Parallel and distributed tools 1998:48-59.

[60] Chung SW, Suh H. DRACO: optimized CC-NUMA system with novel dual-link interconnections to reduce the memory latency. Workshop on MEmory performance 2004:10-16.

[61] Detlefs D, Russell K. Eliminating synchronization-related atomic operations with biased locking and bulk rebiasing. Proceedings of the 21st annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications 2006;41(10):263-272.

[62] Chapin J, Devine S, Gupta A, Lahiri T, Rosenblum M, Teodosiu D. Implementing efficient fault containment for multiprocessors: confining faults in a shared-memory multiprocessor environment. Communications of the ACM 1996;39(9):52-61.

[63] Goodman JR, Kägi A, Rajwar R. Inferential queueing and speculative push for reducing critical communication latencies. Proceedings of the 17th annual international conference on Supercomputing 2003:273-284.

[64] Tsigas P, Zhang Y. Integrating non-blocking synchronisation in parallel applications: performance advantages and methodologies. Proceedings of the 3rd international workshop on Software and performance 2002:55-67.

[65] Cain HW, Lipasti MH. Memory Ordering: A Value-Based Approach. Proceedings of the 31st annual international symposium on Computer architecture 2004:90.

[66] Azimi R, Bilas A. miNI: reducing network interface memory requirements with dynamic handle lookup. Proceedings of the 17th annual international conference on Supercomputing 2003:261-272.

[67] Hoffmann R, Korch M, Rauber T. Performance Evaluation of Task Pools Based on Hardware Synchronization. Proceedings of the 2004 ACM/IEEE conference on Supercomputing 2004:44.

[68] Rinard M, Salcianu A. Pointer and escape analysis for multithreaded programs. Proceedings of the eighth ACM SIGPLAN symposium on Principles and practices of parallel programming 2001:12-23.

[69] Li L, Srinivasan S, Vijaykrishnan N. Simultaneous Partitioning and Frequency Assignment for On-Chip Bus Architectures. Proceedings of the conference on Design, Automation and Test in Europe 2005:218-223.

[70] Keleher PJ. Symmetry and performance in consistency protocols. Proceedings of the 13th international conference on Supercomputing 1999:43-50.

[71] Campenhout JV, Dambre J, Heirman W. Synthetic traffic generation as a tool for dynamic interconnect evaluation. Proceedings of the international workshop on System level interconnect prediction 2007:65-72.

[72] Dahlgren F, Ekman M, Stenström P. TLB and snoop energy-reduction using virtual caches in low-power chip-multiprocessors. Proceedings of the international symposium on Low power electronics and design 2002:243-246.