

# **A Framework for Academic Electronic Journal Publications**

Éamonn Kennedy

A dissertation submitted to the University of Dublin,  
in partial fulfilment of the requirements for the degree of  
Master of Science in Computer Science

September 2003

## **Declaration**

I declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed: \_\_\_\_\_

Éamonn Kennedy

Monday, September 15, 2003

## **Permission to lend and/or copy**

I agree that Trinity College Library may lend or copy this dissertation upon request.

Signed: \_\_\_\_\_

Éamonn Kennedy

Monday, September 15, 2003

## **Acknowledgements**

I would like to thank my supervisor, Mads Haahr, for his invaluable input and guidance throughout the course of this project. I also want to offer my sincerest thanks to my mother and sister for their unyielding love and support throughout this year, without which, I would have given up long ago. Finally, I would like to earnestly thank my classmates, who have both challenged and helped me over the last few months, while also making the whole experience a thoroughly enjoyable one.

## **Abstract**

Academic journals are the backbone of scholarly communication and the preservation of knowledge. For years they have served as a record of progression and an archive of academic debate. They are the medium through which new discoveries become tenets of science, and burgeoning ideas develop into contemporary knowledge.

Unfortunately, an increase in both production and distribution costs imposed by commercial publishers over the last decade, has begun to stifle the ideological purpose of journals. Too often libraries are finding themselves having to settle on a select number of journals that they can subscribe to because of these rises in costs. This leads to an obvious suppression of the natural flow of knowledge through academia.

The processes behind the creation of an academic journal have yet to be standardised, and, as such, journals can subscribe to a variety of different approaches. Journals are most commonly comprised through peer review of potential articles, thus guaranteeing the academic validity of the final publication. However, this is a long and drawn out process for many organisations. New communication technologies have the capacity to make the interaction between editors, reviewers and authors more efficient, thus accelerating the objective of the dissemination of modern knowledge.

The introduction of electronic journals has been seen as the solution to these problems. However, until now, there has been no standardised framework from which organisations can freely explore and develop this option. The following dissertation presents a generic and flexible model of the peer review process, and describes the design and implementation of such a framework. The system created provides a generic, open source solution to creating, managing and maintaining a peer-review academic electronic journal. Also included, is a review of the technological initiatives that have affected the area of electronic journal publication.

# Table of Contents

<b>1</b>	<b>INTRODUCTION</b> .....	<b>1</b>
1.1	BACKGROUND.....	1
1.2	PROJECT GOALS .....	2
1.3	PROJECT SCOPE.....	2
<b>2</b>	<b>STATE OF THE ART</b> .....	<b>4</b>
2.1	ACADEMIC PUBLISHING AND THE OPEN ACCESS INITIATIVE.....	4
2.2	OPEN ARCHIVES INITIATIVE (OAI).....	6
2.3	OPEN ARCHIVES INITIATIVE PROTOCOL FOR METADATA HARVESTING (OAI-PMH) .....	7
2.4	METADATA .....	10
2.5	THE SEMANTIC WEB: .....	11
2.6	THE SCHOLARLY PUBLISHING AN ACADEMIC RESOURCES COALITION (SPARC).....	14
2.7	EDITORIAL MANAGEMENT SYSTEMS .....	16
<b>3</b>	<b>PEER REVIEW PROCESS</b> .....	<b>19</b>
3.1	SPARC: “DECLARING INDEPENDENCE” AND “GAINING INDEPENDENCE” [18][19] .....	19
3.2	ACM: “THE ACM ELECTRONIC PUBLISHING PLAN” [34] .....	20
3.3	IEEE: “INFORMATION FOR AUTHORS” [36] .....	21
3.4	PEER REVIEW PUBLICATION PROCESS UML WORKFLOW MODEL .....	22
3.5	WALKTHROUGH EXPLANATION .....	25
3.6	DECISION POINTS IN THE PROCESS .....	29
3.7	STORAGE CATEGORIES.....	31
3.8	AUTOMATION IN THE PROCESS.....	32
3.9	EVOLUTION TO A DISTRIBUTED SYSTEM.....	33
<b>4</b>	<b>DESIGN</b> .....	<b>35</b>
4.1	DEPLOYMENT DIAGRAM .....	35
4.2	DATA STRUCTURE DESIGN.....	36
4.2.1	Database Design.....	36
4.2.2	File Directory System .....	41
4.3	USE CASE REALIZATIONS.....	41
4.3.1	Administrator/ Editor Use Cases.....	42

4.3.2	Author Use Cases.....	43
4.3.3	Reviewer Use Cases.....	43
4.4	ARCHITECTURAL DESIGN.....	44
4.4.1	Administrator/ Editor System .....	45
4.4.2	Author System.....	67
4.4.3	Reviewer System.....	74
<b>5</b>	<b>IMPLEMENTATION.....</b>	<b>78</b>
5.1	SERVER TECHNOLOGIES.....	78
5.2	FILE TYPES.....	78
5.3	POLICIES .....	79
5.4	AUTOMATION.....	83
<b>6</b>	<b>EVALUATION.....</b>	<b>85</b>
6.1	MODELLING .....	85
6.2	FRAMEWORK.....	85
6.2.1	Creating.....	85
6.2.2	Managing .....	86
6.2.3	Maintaining .....	87
6.3	ADOPTABILITY .....	87
6.4	CROSS PLATFORM ISSUES .....	88
<b>7</b>	<b>CONCLUSIONS.....</b>	<b>89</b>
7.1	GOALS ACHIEVED .....	89
7.2	LIMITATIONS OF RESEARCH .....	89
7.3	RECOMMENDATIONS FOR FUTURE RESEARCH.....	90
<b>8</b>	<b>APPENDICES .....</b>	<b>91</b>
8.1	APPENDIX A: ADMINISTRATOR/ EDITOR SYSTEM .....	91
8.2	APPENDIX B: AUTHOR SYSTEM.....	93
8.3	APPENDIX C: REVIEWER SYSTEM.....	95
<b>9</b>	<b>BIBLIOGRAPHY:.....</b>	<b>97</b>

## Table of Figures

Figure 2.1 Architecture of the OAI-PMH.....	8
Figure 2.2. XML encoding of an OAI-PMH record .....	9
Figure 2.3 Total Number of sites registered to the OAI-PMH [11] .....	11
Figure 2.4 The Percentage of Authors Submitting Online to the Espere System [29].....	18
Figure 3.1. UML Workflow Diagram (1of2).....	23
Figure 3.2. UML Workflow Diagram (2of2).....	24
Figure 3.3. Storage Categories.....	31
Figure 4.1 Framework Deployment Diagram .....	35
Figure 4.2. Database Design .....	36
Figure 4.3. Editor Use Cases.....	42
Figure 4.4. Author Use Cases .....	43
Figure 4.5. Reviewer Use Cases .....	43
Figure 4.6. Administrator System.....	45
Figure 4.7. Author System.....	67
Figure 4.8. Reviewer System.....	74
Figure 5.1 Sample Workflow Diagram.....	81
Figure 5.2. Swim-lane Interaction Diagram.....	82
Figure 8.1 Administrator Welcome .....	91
Figure 8.2 Reviews Received Area.....	91
Figure 8.3. Edit Reviewers.....	92
Figure 8.4 Control Policies .....	92
Figure 8.5. Author Account .....	93
Figure 8.6. Edit Profile.....	94
Figure 8.7 View Past Submissions.....	94
Figure 8.8 Upload Submission.....	95
Figure 8.9. Reviewer Account .....	95
Figure 8.10. Reviewer Edit Profile .....	96



# 1 INTRODUCTION

## 1.1 Background

This thesis describes the work carried out to produce a framework for the electronic journal publication process. The project was proposed by Mads Haahr, and was undertaken to fulfil the requirements for a M.Sc. degree in Computer Science at Trinity College Dublin.

For decades, the area of academic publishing has been a corner stone for the dissemination of knowledge through the academic community and beyond. An academic journal is the poster board for every academic discipline, a yardstick from which progress in a given field can be evaluated. For every branch of learning there is a dedicated academic journal, from archaeology to zoology, academics have dedicated their time to monitor and record the advances in their field. They act as both forums for debate and archives of development. At the heart of this movement is the concept of peer review. This means that in order for an article to be published the leading members in the relevant sphere of study must first assess it against the pertinent criteria for the given field. Once an article is published it becomes a part of the history of a discipline.

For years journals were published in printed form only, however, with the birth of the Internet and electronic archiving techniques, a new hybrid form of journal began to emerge, the electronic academic journal. An electronic academic journal, in simple terms, is an academic journal that is available in an electronic format, be it on the web, or in some electronic file format.

Unfortunately, while the means of production have evolved from hard cover periodicals to searchable databases of articles, the methodologies behind the production have been relatively impervious to change. Organizations still rely on an archaic system of filling hard copies of submitted articles and depending on editors and reviewers alike to keep track of, what can be, an extremely complex paper trail of each journal submission. In order for this integral component of academia to progress and keep in touch with the ever-changing world

outside of academe, work needs to be done to build foundations around which the process of developing a journal can evolve in conjunction with the advances we have seen in publication media.

While there have been advances in technological solutions to the archival and presentation needs of academic journals (which are covered in chapter 2), this researcher believes that there is a need for a freely available technological solution to the management of the peer review process, which needs to be addressed by the academic community.

The intention of this project is to provide a freely accessible, flexible and extensible framework for organizations that wish to further develop their peer review process. The framework created will aid the automation of previously tedious tasks and help cut down on time spent on creating, managing and maintaining an electronic academic journal. Thus increasing the amount of available time for research and evaluation of work within a field. Also, it is hoped that the framework will aid journals move from a hardcopy printed publication, to a more accessible electronic journal.

## **1.2 Project Goals**

The main goal of this project is to highlight the problems and considerations that arise from a peer review approach to electronic publishing, with a specific focus on scholarly journals. Once highlighted, research into methodologies and technologies will unearth means to deal with the constraints involved. With these methodologies in mind, a model of the peer review publication process will be created, taking into account all of the permutations that can arise between different disciplines and organizations. Once modelled, a generic toolkit/framework that will aid in the creation, management and extension of any academic electronic journal will be developed.

## **1.3 Project Scope**

This project will remain focused solely on creating a framework for the electronic publication process. Concerns regarding the presentation of material in a journal, and/ or the

management of journal archives will be touched upon in the research that follows. However, the further development of solutions to these concerns is outside the scope of this project.

## **2 STATE OF THE ART**

The following chapter gives an overview of initiatives that are currently being developed to help deal with the ongoing concerns relating to the propagation of knowledge throughout the academic community. All of the following initiatives discussed have impacted the development of academic electronic journals; yet have been born out of the needs of a variety of different organizations. This is mainly due to the wide influence that electronic publications have on our society as a whole. The organizations that will be focused on are the “Open Access Initiative,” the “Open Archives Initiative,” the “World Wide Web Consortium,” the “Scholarly Publishing and Academic Resources Coalition,” and the “Dublin Core Metadata Initiative.”

It is hoped that this chapter will give an insight into methodologies that currently exist (which could work hand in hand with framework developed in this dissertation) in both archiving and presenting material in an electronic journal.

### **2.1 Academic Publishing and the Open Access Initiative**

In tracing the origins of what is now described as the “Open Access Initiative,” one finds themselves reading about a Hungarian man named George Soros and his social visions of collections of what he termed “Open Societies.” George Soros was born in Hungary, and immigrated at a young age to England where he progressed to graduate from the London School of Economics in 1956. He has been a constant economic and political theorist throughout the years, stamping his opinions through a series of academic articles and books. One of his main agendas throughout his career concerned the promotion of what he termed “open societies.” In relation to academia, an open society holds the diffusion of knowledge, freely and openly in the highest regard. With this goal in mind Soros established his own “Foundation Network,” which to this day seeks...

“ to build free and open societies by supporting an array of activities dealing with the arts and culture; the strengthening of civil society; economic reform; education at all levels;

human rights; legal reform and public administration; media and communications; and public health” [1]

From this organization, emerged the “Open Society Institute” (OSI) [2] and its satellite organizations, the OSI Initiatives. To date, there are over 20 different OSI initiatives addressing a wide variety of concerns from women’s rights to public health reform. They are situated in 29 countries stretching around the world from Albania to Kazakhstan.

All of this leads us on to the 1st of December, 2001 and a revolutionary meeting that occurred in Budapest. Education and free movement of knowledge had always been a large aspect of OSI’s mission. Thus, it was no surprise that they convened the Budapest Open Access Initiative [3]. At this convention scholars and leaders in the academic publishing community joined to devise a strategy through which open access initiatives could join together to share their knowledge and resources to create a structure for a more economically self-sustaining and interoperable academic open access community.

The Budapest Open Access Initiative culminated in the publishing of their “Initiative” document, which outlines, agreed upon strategies, and makes declarations concerning the future of the OAI. The aim of this document being that others will add their signatures and/ or the signatures of their organisations to the initiative in a declaration to abide by the tenets described within [4].

Presently, 2,892 individuals and 212 organisations have signed this document, which clearly shows that there is a growing community that are currently working with these guidelines in mind. While one must admit that the movement is still in its infancy, the new development of the “Directory of Open Access Journals” shows that the movement is growing in the right direction. The “Directory of Open Access Journals” (DOAJ) [5] was launched on the 12<sup>th</sup> of May, 2003. Its mission is to provide a searchable directory of open access journals from a variety of different academic disciplines. They define an “Open Access Journal” as one that uses “a funding model that does not charge readers or their institutions for access.” [6]

The DOAJ was set up and developed in reaction to the OAI-Budapest, by Lund University Libraries. It is both supported and funded by the OAI-Budapest and also has an affiliation with “The Scholarly Publishing and Academic Resources Coalition” (SPARC), which the researcher will cover later in this document. In relation to academic electronic

journals, it is one of the most concise listings of academic journals on the web. One of the more interesting aspects of this directory of journals is that not only does it conform to the guidelines set out by the OAI-Budapest but it also supports the Open Archives Initiative protocol for Metadata Harvesting (OAI-PMH) [7], again a topic that the researcher will cover later in this document.

## 2.2 Open Archives Initiative (OAI)

In October 1999, two years before the OAI-Budapest, the Open Archives Committee met for the first time in Santa Fe, Mexico, to devise a system to aid the proliferation of data from e-documents, using defined archival frameworks. The eventual link between both the Open Access Initiative and the Open Archives Initiative, is a useful example of how standardizing processes and frameworks within two separate disciplines can lead to larger and more effective projects, *e.g.* The Directory of Open Access Journals.

As stated, the Open Archives Initiative was born out of what is now termed the “Santa Fe Convention.” [8] The convention was a historic event for academic e-publishing, in that it brought together organisations ranging from NASA to MIT, and over two days they developed a plan of action that they all agreed to endorse. The convention was sponsored by the “Council on Library and Information Resources” (CLIR), Digital Library Federation (DLF), SPARC and the “Association of Research Libraries” (ARL), all of which had been involved in the movement towards a common goal of sharing out scholarly archival information. The core document of the convention itself can still be accessed [9], however the peripheral effect of the convention, *i.e.* formation of the OAI, created a much greater impression on the future of electronic archiving.

The core document mainly outlined ways in which an archive could comply with the key guidelines set out by the convention. In brief, these guidelines outlined what constitutes an e-print archive, how records should be named using persistent identifiers, the definition of a data provider and a service provider, and introduced the ideas of metadata and metadata harvesting (concepts that will be discussed later in this document). Historically speaking, the most important aspect of the guidelines set out by the convention was that they clearly and unambiguously defined how an electronic archive could be used to its full potential, while

also defining a vocabulary that would eventually be used in the creation of the “Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH).”

After the convention, a “Steering Committee” was set up to insure that the movement would continue to evolve. The Steering Committee mainly consists of a group of representatives from the original organisations that were involved in the convention [10]. Since the original convention more workshops have been set up to help extend and promote the OAI, until eventually the Santa Fe convention evolved into an organisation dedicated to developing the OAI-PMH.

## **2.3 Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)**

As stated, the OAI-PMH grew out of the interest generated by the Santa Fe convention. It began as a purely experimental project, targeting early adapters in the field with a view to an eventual wide scale adoption by libraries and academic archives. The protocol itself is built upon the assumption that two types of users will exploit its power. The OAI describes its users as either “Data Providers” or “Service Providers [11].”

Data providers are the users that actually create and archive the material that others may want to access. They do this by using the OAI-PMH to expose specific metadata about their resources/ materials.

Service Providers are the users, which will use the protocol to “harvest”/ gather this metadata and possibly build further applications on top of this new harvested material.

The following diagram outlines the how the protocol is used to support the exchange of archived data between a Data provider and a Service Provider. The Data provider holds a collection, or repository, of records of metadata on the information it has available in its archives. Service providers can then use a harvester to query the repository using an OAI-PMH request over HTTP. Once the repository receives the request, it then returns an XML encoded byte stream to return the relevant records to the request. Eventually the service provider can utilise the records it has collected to build higher-level applications on top of the information it has collected.

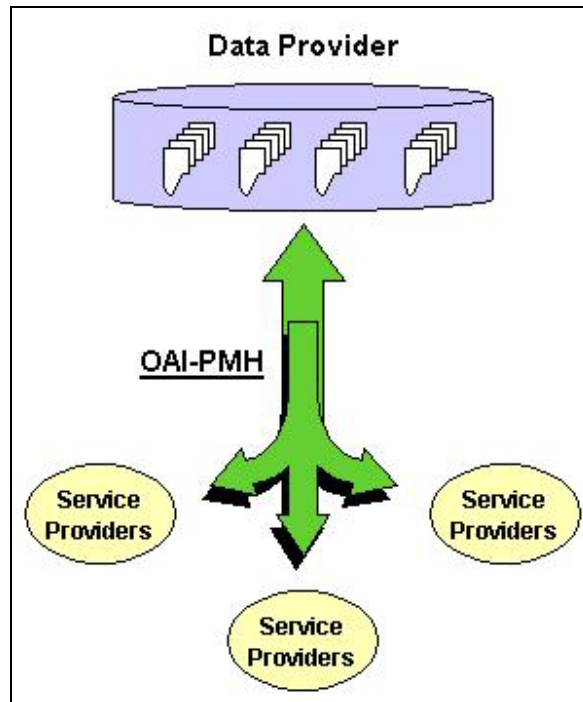


Figure 2.1 Architecture of the OAI-PMH

All of the information transmitted between the data providers and service providers occurs using XML over HTTP.

An example of the contents of an XML encoding of a records follows:



```

<header>
  <identifier>oai:arXiv.org:cs/0112017</identifier>
  <timestamp>2002-02-28</timestamp>
  <setSpec>cs</setSpec>
  <setSpec>math</setSpec>
</header>
<metadata>
  <oai_dc:dc
    xmlns:oai_dc="http://www.openarchives.org/OAI/2.0/oai_dc/"
    xmlns:dc="http://purl.org/dc/elements/1.1/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/oai_dc/
      http://www.openarchives.org/OAI/2.0/oai_dc.xsd">
    <dc:title>Using Structural Metadata to Localize Experience of Digital
      Content</dc:title>
    <dc:creator>Dushay, Naomi</dc:creator>
    <dc:subject>Digital Libraries</dc:subject>
    <dc:description>With the increasing technical sophistication of both
      information consumers and providers, there is increasing demand for
      more meaningful experiences of digital information. We present a
      framework that separates digital object experience, or rendering,
      from digital object storage and manipulation, so the
      rendering can be tailored to particular communities of users.
    </dc:description>
    <dc:description>Comment: 23 pages including 2 appendices,
      8 figures</dc:description>
    <dc:date>2001-12-14</dc:date>
    <dc:type>e-print</dc:type>
    <dc:identifier>http://arXiv.org/abs/cs/0112017</dc:identifier>
  </oai_dc:dc>
</metadata>
<about>
  <provenance
    xmlns="http://www.openarchives.org/OAI/2.0/provenance"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.openarchives.org/OAI/2.0/provenance
      http://www.openarchives.org/OAI/2.0/provenance.xsd">
    <originDescription harvestDate="2002-02-02T14:10:02Z" altered="true">
    <baseURL>http://the.oa.org</baseURL>
    <identifier>oai:r2.org:klik001</identifier>
    <timestamp>2002-01-01</timestamp>
    <metadataNamespace>http://www.openarchives.org/OAI/2.0/oai_dc</metadataNamespace>
    </originDescription>
  </provenance>
</about>

```

Figure 2.2. XML encoding of an OAI-PMH record

## 2.4 Metadata

In simple terms, metadata is information about information, or information that describes a resource, similar to an abstract for a paper. There has been much research into the area of metadata. However the Dublin Core Metadata Initiative (DCMI) is widely accepted as the leader in the field.

The Dublin Core Metadata Initiative is an organisation that has its history in a workshop held in Dublin, Ohio in 1995 [12]. Since that workshop, it has held up to nine similar workshops around the world. The initiative was set up to promote the development and adoption of metadata standards. In relation to the OAI, the OAI-PMH works in conjunction with the standards that the DCMI has set out for describing metadata resource elements. The OAI-PMH works on a framework in which data providers must represent their metadata in accordance with the DCMI [13]. Thus, all Data Providers and Service Providers will be exchanging information in the same format, allowing for the automation of information retrieval and management.

An illustrative narrative of how the OAI-PMH can work in an academic environment follows: An academic journal has an archive/ repository of previously published material. The organisation configures its archive in accordance with the OAI-PMH as a Data Provider (An important note on the technologies used by the OAI-PMH is that they are all open source). Once they create this archive and the metadata that accompanies it, a Service Provider, *e.g.* a library, could easily catalogue their entire archive and combine the metadata from the journals archive with other similar resources. Finally, the library could set up an easily searchable resource of articles from many different journals, as a service to students/ researchers.

A report published by the OAI in 2002, [11] states that by May 2002 there were approximately 6 million metadata records that could be harvested by the OAI-PMH. Seeing that users of the protocol do not necessarily have to register with the OAI, it is difficult to predict the number of users utilizing the protocol. However, while the report admits that it is difficult to collect metrics that would measure the success of the protocol, it purports that there is anecdotal evidence to suggest that the number of registered sites represent less than half of the actual implementers of the OAI-PMH.

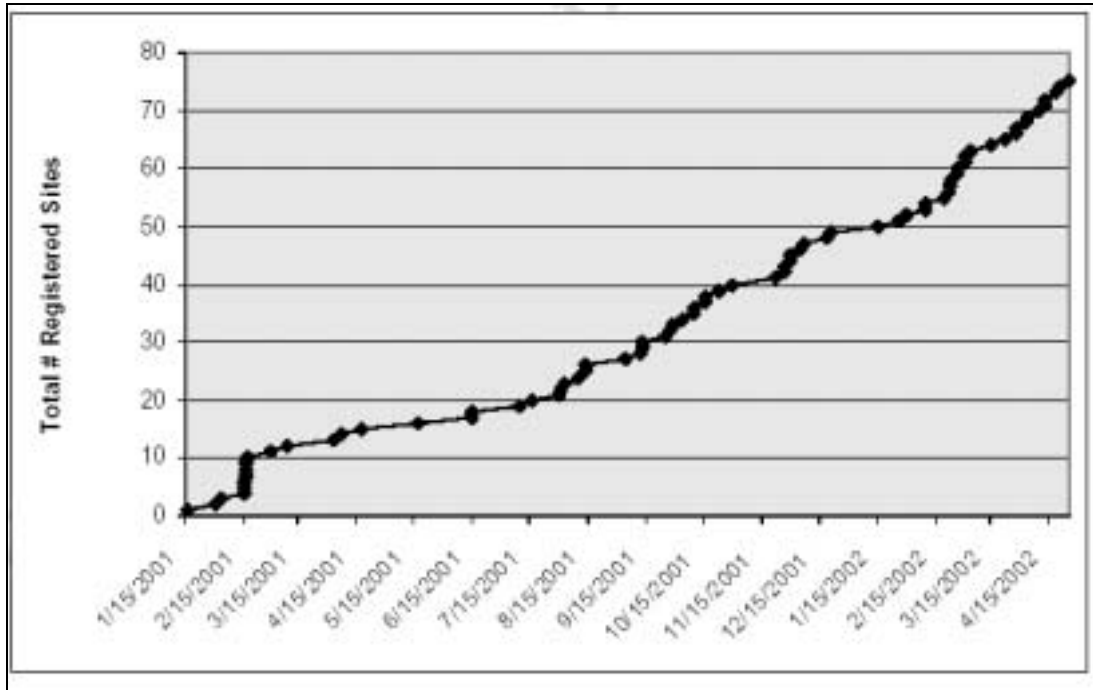


Figure 2.3 Total Number of sites registered to the OAI-PMH [11]

When mentioning the DCMI and the subject of the formatting of data for the purpose of automating processes to harvest information from that data, the W3Cs Semantic Web cannot be ignored.

## 2.5 The Semantic Web:

The semantic web has its origin in the writings of Tim Berners-Lee. In his own words: “The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation.” [14].

The original World Wide Web was built for human consumption, therefore the representation of information was designed to be read and interpreted by humans. This in turn leads to many implicit implications about the format of information on the web, *i.e.* humans can easily distinguish semantic differences based on context, *e.g.* if we read the word “stock” on a “CNN’s Money” site, not many people would get confused between whether the word referred to a cooking broth or a financial stock. However, with the massive rise of automated web crawlers and intelligent agents which search through web information trying to decipher

data that lies in different web resources, it quickly becomes obvious that the formatting rules of the web do not allow for semantics on information to be easily linked with the information itself. Enter, the semantic web. Basically, it is a metadata layer that sits abstractly on top of the web, where information can be labelled with unambiguous machine-readable descriptors. There are four building blocks that create the semantic web; Uniform Resource Identifiers (URIs), XML, The Resource Description Framework (RDF) and XML Schemas/ Ontologies.

Uniform Resource Identifiers are ways of identifying the location or meaning of a resource, *e.g.* a URL is a form of URI. The current problem with URIs is that anyone can create a URI for anything they want, *i.e.* a URI only provides an identifier for a resource, and it does not imply any semantic meaning. However, if used in conjunction with XML and some form of standardization, a URI could hold semantic information.

The semantic web uses URIs represented in XML to mark up different elements of resources. The best and most effective way of explaining this is through an example:

Lets examine the sentence, “I got a new pet dog.” In its current format, if a computer program was to process the sentence, its semantic meaning could easily be lost, *i.e.* it would have little clue as to meaning of what, or whom “I” was referring to.

However, if we marked it up using XML:

```
<sentence>
  <person href="http://Eamonn.com/"> I </person>
  got a new pet
  <animal>dog</animal>.
</sentence>
```

Quickly, we can imagine how a program could decipher semantic information form the XML representation of the sentence. While the utilization of such a representation is clear, the matter of standardizing such representations still needs to be addressed. This is where the W3Cs Resource Description Framework (RDF) fits in to the semantic web.

The W3Cs RDF Primer document <sup>[15]</sup> outlines a means of formatting XML documents for use in the semantic web. The goal of this formatting technique is to apply a

standard on XML documents that would make statements easier for machines to process. In brief, every RDF statement has three parts to it; a subject, a predicate (which relates the subject to the object) and an object. Again, an example of these concepts in practice, is the best way to explain them.

Consider the sentence, “I really like ‘Weaving the Web’.” In a similar fashion to the last example, it would be difficult for a machine to gain any semantic meaning from the sentence. However, the official RDF representation in XML adds a certain degree of clarification.

```
<rdf:RDF xmlns:rdf=http://www.w3.org/1999/02/22-rdf-syntax-ns#
  xmlns:love="http://love.example.org/terms/" >

  <rdf:Description rdf:about="http://Eamonn.com/">
    <love:reallyLikes
      rdf:resource="http://www.w3.org/People/BernersLee/Weaving"/>
  </rdf:Description>

</rdf:RDF>
```

In the above example, the researcher acts as the subject. The object of the RDF statement is the book “Weaving the Web” and the predicate that joins the subject to the book is the term “like.” With this standardisation in practice, the means through which a program could be written to interpret the information in the given format becomes apparent.

The next step of adding usable semantics to a resource lies in using XML Schemas/ Ontologies. A schema and an ontology are ways to describe the meaning and the relationships of terms, in other words they help us describe the resources identified by URIs. Presently there are two accepted forms of schema/ ontology; RDF Schemas and DAML+OIL (Darpa Agent Mark-up Language with Ontology Interface Layer). Also, the W3C have started a Web Ontology (WebOnt). Basically, these schemas and ontologies provide a vocabulary to indicate that resources are referring to specific kinds of classes of information.

To extend a previous example, *e.g.* a dog is a type of animal, we could create an RDF schema to provide us with information on different types of animals. We would begin by declaring an “animal” class.

```
ex:Animal rdf:type rdfs:Class.
```

(where “ex” stands for a namespace URIref  
e.g. <http://www.example.org/schemas/animal>)

Next we could declare a resource “dog” as being a subset of the class “animal.”

```
ex:dog rdf:subClassOf ex:animal.
```

Hopefully the example above illustrates how useful the schemas can be in adding semantics to resources, which in turn could increase the possible utility of any academic electronic journal.

## **2.6 The Scholarly Publishing an Academic Resources Coalition (SPARC)**

So far, this chapter has outlined the major initiatives that have impacted the archiving and presentation of information in academic electronic journals. The next section will discuss the concerns of creating, managing and the actual implementation processes involved in the production of an academic electronic journal.

In the past twenty years, many would argue that the realm of academic publishing has gone through an ideological crisis. The publication of scholarly work is at the heart of academia. It is the driving force behind the wide dissemination of knowledge that has occurred in the last century. However, the means through which this transpires has been widely debated. Most scholarly publications are maintained and produced within the academic community. The articles are written, reviewed, edited and aimed at scholars within each discipline. However, the medium through which this transfer of knowledge takes place has increasingly become under the control of large publishing companies. Once an academic journal is produced by an organization, they often outsource the printing and distribution to commercial publication company. As this process becomes more common, there is a knock-on effect of an increase in the cost prices of academic journals. Indeed,

“Data collected by the Association of Research Libraries (ARL) (2001), a membership organization of over 120 of the largest research libraries in North America, reveal that the unit cost paid by research libraries for serials increased by 226% between 1986 and 2000. (In comparison, over the same time period, the consumer price index increased by 57%.)” [16]

With the increase in the pricing of journals, fewer libraries can afford to stock every journal available to their students. In practical terms this has the effect of decreasing the variety of journals available in any given library, and decreases the spread of information throughout the academic community.

This in turn, led to the formation of SPARC by the ARL, in 1997. The mission of the organization was to act as a “catalyst for change through the creation of a more competitive marketplace for research information” [16]. In concrete terms, the organisation wanted to give power back to the source of scholarly publications, *i.e.* the scholars themselves.

Through movements such as “Create Change,<sup>[17]</sup>” “Declaring Independence,<sup>[18]</sup>” and “Gaining Independence<sup>[19]</sup>,” SPARC has encouraged academic organisations to take control of their publication process. They invite both libraries and academic publishing organisations to develop strategies to take back the power from large publishing companies, with their theories being very much in line with the “Open Access Initiative.” These initiatives will be covered later in this document.

One of the more interesting factions of SPARC is their “Alternative Program” [20]. The aim of this program is to provide editors and authors with practical alternatives for publishing, lessening their dependence on “for-profit” publishing companies. In relation to this project, an interesting note on the Alternatives Program, is that in order for an organization to gain membership, they must prove that they are using “electronic media or technology to reduce cost or otherwise obtain competitive advantage and/ or value for user [20].” Using the framework created in this project would clearly aid an organisation in gaining membership to SPARC’s Alternative Program.

Presently, there are many working examples of journals that subscribe to the mission set out by SPARC, a list of its members can be found here, [21]. Also, it could be argued that most of the sites listed by the DOAJ have similar objectives as those set out by SPARC.

While SPARC is obviously one of the leaders in setting out guidelines for academic journals, its guides lack a major component; how to technically implement an electronic journal? SPARC's guidelines are unquestionably helpful from an economic perspective, but in the researcher's opinion, they fail considerably on technical directives. Many start-up journals may opt to publish their work simply as a paper-based journal, which one might argue is an effective choice and easy to implement. Unfortunately this can be a costly option. However, if one were to publish in accordance with the "open access" directives, an electronic copy of the journal would aid in a further reaching dissemination of the material. Furthermore, if the technology were available to automate the entire publication process, (*i.e.* from receiving a submission to publishing that submission as an article) why not take advantage of said technology? Academia should be an environment where new technological developments can be harnessed and tested, a sphere where one can embrace progress rather than ignore it.

With regard to producing an online copy of a journal, there are many tutorials available on the web, which could help the less technically minded set up and manage an online edition of their journal. There is software available free of charge which most people could exploit to develop an electronic copy of their journal. However, when one searches for an electronic solution, to aid in the management of the processes behind the actual publication of material (*i.e.* the submission and peer review process), there is a serious lack of resources. The following section summarizes the few available options for managing an academic electronic journal.

## **2.7 Editorial Management Systems**

In April 2001, "The Association of Learned and Professional Society Publishers" (ALPSP)<sup>[22]</sup>, a journal dedicated to strengthening the community of non-profit publishers, published an article written by Dee Wood concerning electronic peer review systems<sup>[23]</sup>. The article examined the availability of systems that could aid a non-profit journal in managing its peer review process. The crux of the article is a review of the leading systems available to publishers at that point in time. The author compiled an in depth comparison of commercial solutions, which breaks down the major aspects of the process and comments on how each system handled them. The systems reviewed were: "ESPERE,"<sup>[24]</sup> "Manuscript Central,"<sup>[25]</sup>



“PaperPath,” “Rapid Review” [26] and “XpressTrack” [27].” Each of these systems could either host the software needed for managing the processes involved and make them available for use online, or else could make the software available for use on a local installation. Unfortunately, of the five systems reviewed, only four remain today. “PaperPath” seems to no longer exist, as all links referencing the company are presently broken, and the site [www.paperpath.com](http://www.paperpath.com) no longer exists. The other sites still report to be providing a service.

One cannot argue underestimate the usefulness of these solutions and the degree to which they could alleviate, what could be, an arduous undertaking. However, an aspect of these services that was of great interest, especially when looked at from an ‘open access’ perspective, was that each solution carried a subscription fee with it. Three of the four sites, namely “Manuscript Central,” “Rapid Review” and “XpressTrack,” are clearly commercial ventures. They were established on the premise that the solutions they provide would be profitable for the companies themselves. Simply, they charge a fee, provide a service and reap the profits accordingly. However, what most striking about the “Espere” system was that it was set up by the ALPSP, which is itself a not-for-profit organisation.

The administrator of the “Espere” system provided this researcher with access to their system, offering access to walk through the services they provide. Permission was granted to test out the both the administrator’s section of the site, and the author/ submitter section. While running through the variety of options for both administrators and authors that the system provides, it is easy to see that the founders of the system truly understood the mechanisms involved in a peer review process. The support that the system provides is both, flexible and succinct in nature, while the interfaces to the system are clear and relatively simple to use. Presently, there are sixteen journals using the system [28], all of which have little relation to computer science.

The system provides a simple solution for peer review journals to automate their processes, without the administrators having to get tied up in the technology behind the implementation. For this purpose, it is difficult to find fault in the “Espere” system. However, one cannot overlook the fact that they charge an annual fee to journals that wish to utilise the system. One might argue that this fee could be regarded as a minimal offering, and that ultimately journals are paying for a working system that will be maintained and updated by professionals. A reflection of how useful the system has been, is shown in Espere’s statistics on author submissions that occurred through the system.

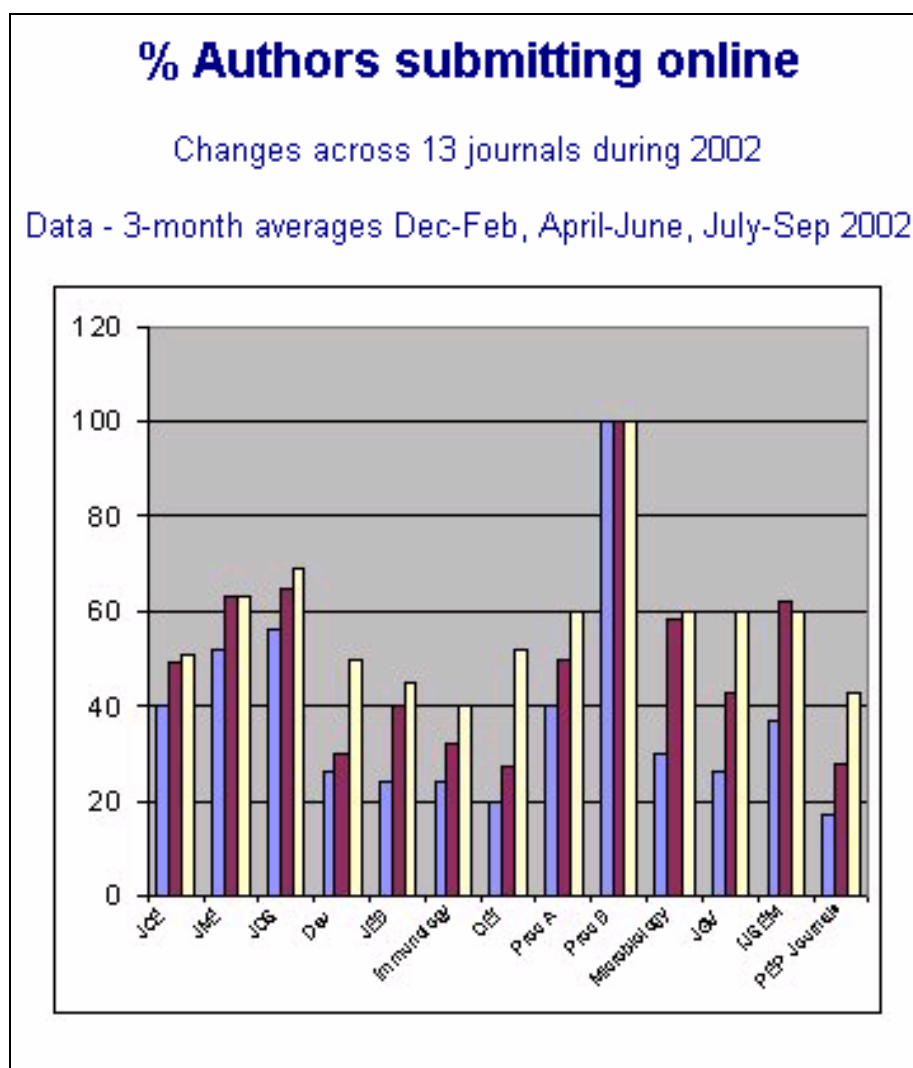


Figure 2.4 The Percentage of Authors Submitting Online to the Espere System [29]

In an age where servers are widely available to academic institutions, there is little excuse for an academic institution to outsource such a service.

### 3 PEER REVIEW PROCESS

While there are many guidelines available outlining factions of the publication process, from journalistic approaches [30] to accessibility issues [31], subject media and stylistic designing of the final product [32], little research could be found relating to the overall process involved in the publication of a peer review academic journal; notwithstanding the processes involved in the production of an electronic academic journal.

What is most surprising about this fact however is that there are literally hundreds of ‘e-journals’ available to the academic community [33]. Therefore, before an attempt to provide a framework through which a peer review electronic academic journal could be produced from, a model of the process/ processes involved needed to be created.

The following chapter outlines the research behind the model created and gives a detailed description of the model itself. In the creation of this model, information was gathered from many different sources; both from personal experience, the experience of the Editor-in-Chief of Crossings Mr. Mads Haahr, and various online sources where possible. Unfortunately, in the same way that no attempts at modelling the process could be found, no organisations were found that clearly outlined their publication process, or articles relating to the specific administration of production. Following is a synopsis of three publication guidelines, retrieved from a group of the more prominent and recognisable organisations involved in electronic publishing.

#### 3.1 SPARC: “Declaring Independence” and “Gaining Independence” [18][19]

SPARC (the Scholarly Publishing and Academic Coalition) in line with TRLN (the Triangle Research Libraries Network) published these two documents to work as guidelines to aid academic organisations ‘gain independence’ from publishing companies. They argue that, “During the last four years, the average cost of a commercially-published journal has risen nearly 50%” [18] creating a knock-on effect where academic publishing is more concerned with becoming a commercially viable enterprise than helping the dissemination of

knowledge in a given field. While these guidelines offer extremely pertinent information regarding the economic and marketing issues surrounding academic publishing, they offer little information relating to the actual review and publication process itself.

### **3.2 ACM: “The ACM Electronic Publishing Plan”<sup>[34]</sup>**

The main aim of this document is to outline the ACM’s (Association for Computer Machinery) vision for the development of the electronic version of their printed journals. The document also raises questions pertaining to their copyright policies <sup>[35]</sup> and how they can be preserved in their future electronic publications. Interestingly, while this plan is outlining the ACM’s electronic publication future, it briefly outlines ‘The Scientific Publishing Tradition.’ According to the ACM, a journal passes through four phases of development and three ‘key moments’ of public declaration. The author describes the four phases of development in a chronological ordering:

1. The preparation of the submission by the author.
2. The review and revision of the submission by the editor and referees/ reviewers.
3. The actual publication of the manuscript, which involves sending it to a publication office for editing and layout, and culminates in the distribution of the manuscript to the readers.
4. Finally, the manuscript is preserved through a process of archiving and indexing the said manuscript.

The author proceeds to describe the three key public declarations as:

1. Declaration of submission by the author.
2. Declaration of acceptance by the editor, usually involving a transfer of copyright

3. The actual publication and distribution of the journal.

Although, this plan did not go as far as modelling the publication process, it helped to outline some key steps/ issues that need to be covered if a model is created.

### **3.3 IEEE: “Information for Authors”** [36]

The purpose of this document is to inform authors of potential submissions of their rights and responsibilities, while also describing the general publication procedures used by the IEEE journals. In a similar fashion to the ACMs Electronic Publishing Plan, the authors of this document provide a solid description of the IEEE copyright policies, while also giving an outline of the steps involved in publishing an IEEE scholarly publication. In the following list, an attempt is made to summarise the main phases of the publication process that the IEEE outline:

1. Initial Decisions: Basically this phase runs through the decisions that need to be taken into account before submitting to an IEEE publication, *i.e.* the submission guidelines, and international intellectual property law.
2. Formats: Three types of submissions are accepted by the IEEE, Transactions (8 to 10 printed pages), Journals (same length approximately as a transaction, except more specialised in nature), and finally Letters (short papers approximately 4 printed pages).
3. Peer Review: Editors decide whether a submission is suitable for their publication. If the submission is considered, the submission is sent to a group of reviewers and enters into a system of review and revision.

4. Final Acceptance: Once a submission gains final acceptance, the author is informed and a copyright form is sent out to them. Along with a copyright form, the author might need to complete other informational forms depending on the publication involved.
5. Preparation of Electronic and Final Manuscripts: This phase surrounds simple formatting of text and images for publication.
6. Author Proofs: The author is sent a proof copy of their soon to be published submission, in order to give them one last opportunity to make any revisions.
7. Printed Issue: The actual publication of the submitted material.

### **3.4 Peer Review Publication Process UML workflow model**

Given the above guidelines, this researcher's approach to modelling the processes involved is, in its nature, very simple. The following model is intended to be as generic as possible. Few assumptions have been made about the organisational structure behind the process, within realistic constraints. Apart from assuming that a publication has some form of centralised decision-making, *i.e.* an editorial board, no assumptions about any organisational hierarchy, or review methodology/ ethics are needed for this model to reflect the activities involved in academic publications. The implications of this approach are that the framework expressed will result in the creation of a malleable and extensible system, which can be easily adapted by any organisation. Also, hopefully it will act as a point of reference around which debate can grow and prosper, in relation to contentious issues involved in the publication of a journal for the sake of the dissemination of knowledge.

The following figure is a standard UML activity diagram describing the academic peer review publication process. Essentially, it is a simple diagram to show the paper trail of the submission of a paper from an author through to the actual publishing of the paper by an organisation. It pinpoints critical stages in the lifecycle of a submission, from the submission itself to either its rejection or eventual publication.

# UML Workflow Diagram (1 of 2)

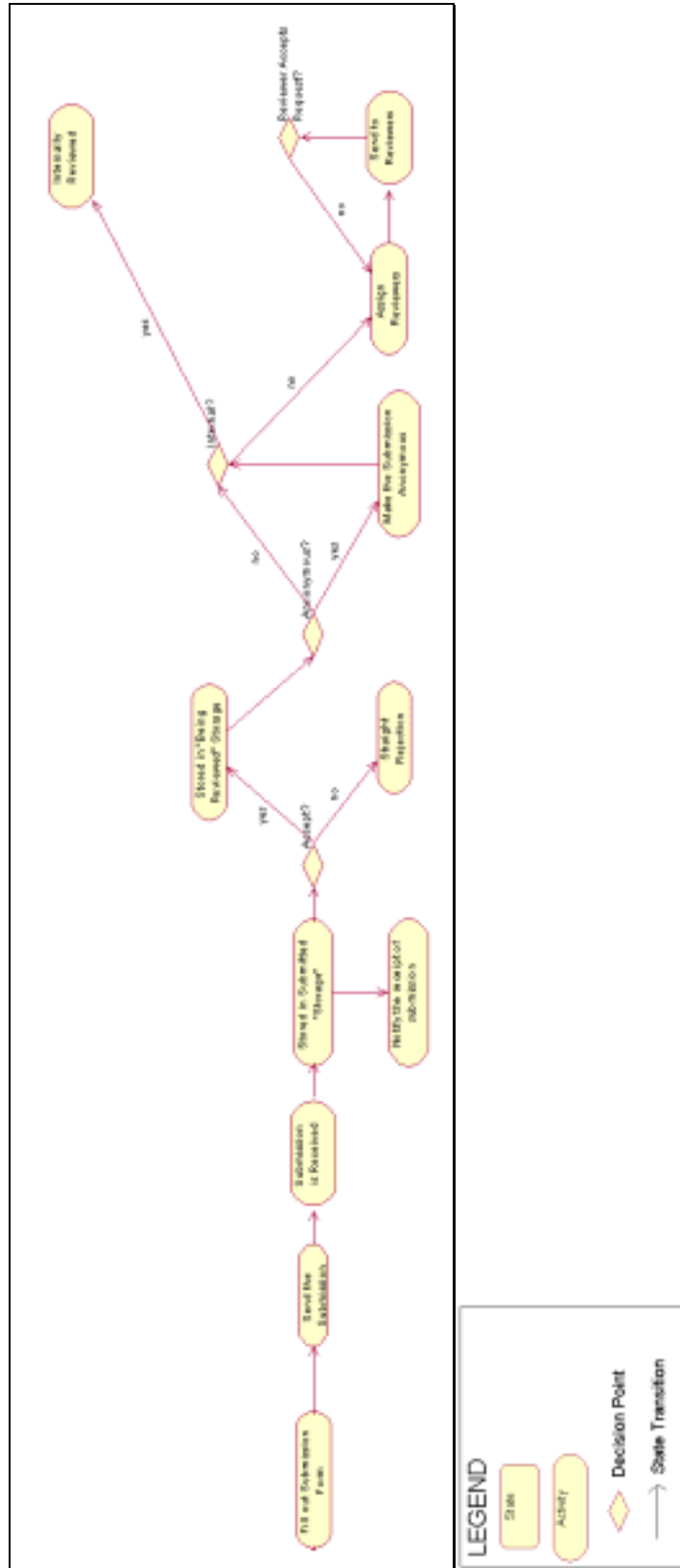


Figure 3.1. UML Workflow Diagram (1of2)





### 3.5 Walkthrough Explanation

1. The Submitter fills out the appropriate forms that are needed by the organisation as accompaniments to all submissions, *i.e.* The title of the submission, affiliations of the submitter etc....
2. The Submitter proceeds to send the paper/ submission along with the form they have completed.
3. The Submission is received by the organisation.
4. In some manner the submission is stored by that organisation. Whether it is stored physically or digitally is up to the organisation.
5. Once the submission is received the organisation normally notifies the Submitter of the receipt of their submission.
6. At this stage a decision must be made as to whether the submission should be rejected straight away, or proceed to the review stage, *i.e.* whether the submission has grounds to be reviewed.
  - a. If rejected the submitter is notified
  - b. Otherwise, Store the submission in a filing system with other papers up for review, *i.e.* In a *Being Reviewed* folder
7. Once the submission has passed the initial acceptance criteria, a decision must be made whether the submission will be treated as coming from an anonymous source or not.

- a. If the submission is to be anonymous, then the paper must be edited in such a way that the author's identity is concealed
8. Next, the organisation must decide on whether the submission should be sent out to external reviewers or reviewed by the organisation's internal editors. This decision normally manifests itself on an organisational level, *i.e.* whether the journal's editorial board is comprised of the reviewers.
  - a. If the reviewing is internal the editorial board can easily handle the reports.
  - b. If the reviewing is external the editorial board must:
    - i. Assign the submission to the reviewers of choice.
    - ii. Send the submissions to the reviewers.
    - iii. If the Reviewer accepts the review request, they will continue to review the submission and report back to the editors. Otherwise, a replacement reviewer might need to be assigned the submission.
9. Once reviewed the submissions can then be stored in a *Reviewed* file storage system, thus indicating the fact that the submission has at this stage been reviewed. Normally, the submission and the reviews written about them are stored together.
10. The editorial board can subsequently read the reviews and thus make a informed decision, regarding the next stage in the life of the submission. Typically the editors will have four options available to them:
  - a. To reject the submission but encourage the submitter to resubmit their work at a later date.

- b. Ask the Submitter to address a particular aspect/ aspects of the submission and accordingly resubmit their work.
  - c. Reject the submission outright.
  - d. Accept the submission in its current form.
11. Normal practice in an academic peer-review process includes returning the reviewers critique to the submitter. At this point in the process, the organisation might need to decide whether they wish to keep the identity of their reviewers anonymous.
12. If the Submitter is asked to resubmit their work, upon resubmission the paper is once again reviewed, to ensure if the matters under contention were addressed.
- a. If they were addressed in a suitable/ appropriate manner, the submission will then move along in the process to the editing phase.
  - b. If the editors decide that another resubmission is required, the submitter will be asked to once again resubmit. The number of times that a Submitter is allowed to resubmit their work is a matter that is normally predefined by the organisation.
  - c. Often the editors might feel that the resubmission grants a further reviewing/ re-review by external reviewers. If so the submission is passed back along the process, to the point of assigning an external reviewer and must proceed once again from that point.
13. Upon the final acceptance of the submission the submission itself must be edited according to the stylistic guidelines/ design of the journal in question. At this stage, where the submission resides within the organisation will change, *i.e.* it will be

moved to a file storage facility indicating that it is presently being edited, for content or stylistic reasons.

14. Once the submission is edited, the organisation might then need to return the final proof/ draft of the submission to the Submitter and ask for their permission to publish it in its current state. The organisation might also at this stage need to ask the Submitter to transfer their copyright privileges to the organisation, or some form of legal assignment of copyright.

a. If the Submitter agrees to allow the organisation to publish their work, the organisation will then place the submission with other works that are going to be published in the next edition of their publication.

b. Otherwise, the paper cannot be published.

15. The final stage is to actually publish the submission through whatever medium the organisation chooses.

### 3.6 Decision points in the process

In order to use this model productively, *i.e.* as a basis upon which a development framework can be conceived, the choices that editors have throughout the process must be highlighted. When looking at the workflow of the process, it is evident that there are seven main ‘decision’ points. These are points in the process when a centralised decision must be made concerning the future of the submitted work.

1. **The Initial Acceptance.** This is the juncture at which the editors will briefly review the submission and pose the following questions:
  - a. Is it a relevant submission, *i.e.* does it relate to the *Raison d’Être* of the journal? *E.g.* Crossings *Raison d’Être* <sup>[37]</sup>
  - b. Does the submission follow the submission guidelines? *e.g.* Crossings Contributor’s Guide <sup>[38]</sup>
2. **Anonymous Reviewing.** This is more of a policy question that relates to the ‘modus operandi’ of the organisation. Some organisations prefer to hide the identity of the submitter from the reviewers, so that the review will be as impartial as possible. While others may feel that reviewer anonymity may result in a move towards destructive rather than constructive criticism of a submission <sup>[39]</sup>. It is a matter that has spawned debate in the past and will continue to encourage further debate.
3. **Internal Reviewing.** Again, this is a policy question. Different organisations will have different modes of reviewing a submission. While some have editorial board meetings, where submissions are reviewed and discussed inside the organisation, others may have a listing of peers in a distinct field that might have a more learned perspective on the subject of a particular submission. Therefore, a choice must be

made whether to keep all reviewing internal, external or sometimes a mixture of both may occur.

4. **Anonymous Reporting.** Some journals policies involve sending a report, often consisting of reviews of the submission, to the submitter after they have made a decision regarding their acceptance of the submission. A decision must be made by the organisation on whether these reviews will include the names of the reviewers, or whether the identity of the reviewers will be hidden from the submitters.
5. **Re-Submission.** Often many organisations may give an opportunity to a submitter to resubmit their paper after taking the opinions of the reviewers into consideration. Again a policy should be in place concerning the number of times a submission can be accepted for resubmission after it was initially rejected for publication. According to a report presented at the 'Electronic Peer Review Internet Conference (1996),' "an estimated 80 per cent of papers are returned for revision" [40].
6. **Second Re-Submission.** If the submission is still found lacking after a resubmission, the editors must decide (either by referring to a 'constitutional' document for their organisation or deciding on a submission-by-submission basis) whether they can accept more than one resubmission.
7. **Copyright.** This is a decision where the submitter is in control of the destiny of their submission. In order for the work of an author to be published by an organisation the organisation might need to have copyright control over the authors work. Sometimes this might be agreed upon as a precondition to submitting any work to the journal, or else, the transfer of copyright might have to occur once the submission gains acceptance status from the organisation to be published [35].

### 3.7 Storage Categories

Below are five distinct categories under which an organisation could store their submissions that would reflect the different stages/ statuses through the lifecycle of a submission through the publication process.



Figure 3.3. Storage Categories

1. **Submitted.** Contains the initial submissions.
2. **Being Reviewed.** Contains the submissions that made it through the initial acceptance stage. Could also contain information pertaining to the identity of the reviewers of each submission (if reviewing is external) and the date that each submission was sent/ checked-out by the reviewers.
3. **Reviewed.** Contains all submissions that have been reviewed, coupled with the reports/ reviews obtained from the reviewers.

4. **Editing.** Contains all submissions that have been accepted for publishing and are currently being edited/ formatted by the editors.
5. **Published.** Contains all current and previously published materials.

### **3.8 Automation in the Process**

One of the major aims when of this project was to automate, where possible, the peer review publication process. When dissecting the workflow model, opportunities for automation, and thus increasing the efficiency of the process become apparent. These opportunities have been separated into 4 different categories:

#### **1. Storage**

A web interface could be used to aid the Editor-in-Chief move a submission through the different storage levels outlined in the model.

#### **2. Policies**

Policies reflecting an organisations stance on policy decisions could be accounted for when setting up a system, *i.e.* the outlined policy decision points could be hardwired, so to speak, at the initial set up phase. Thus, reducing the need for ‘on-the-fly’ centralised decisions being made on such topics as: Submitter Anonymity, Reviewer Anonymity, Internal/ External reviewing, How many times a person can resubmit their work, When the transfer of copyright occurs.

#### **3. Communication**

Communication between the three parties involved in the process, *i.e.* the submitter, the reviewer, and the editor could easily have an automated aspect involved, *e.g.* automated emailing.

There are 4 main areas where I feel that automated communication could occur.



1. Confirmation of the receipt of a submission
2. Sending of the submissions to the reviewers
3. Sending reports to submitters communicating the status of their submissions, *i.e.* straight rejection communications.
4. Asking for transfer of copyright

#### **4. Editing for anonymity**

Based on the formats of both the submissions and the reports, a parser could be created to remove the names of the authors from the text.

### **3.9 Evolution to a distributed system**

“A distributed system is one in which components located at networked computers communicate and coordinate their actions only by passing messages. This definition leads to the following characteristics of distributed systems: concurrency of components, lack of a global clock and independent failures of components.” [41]

Using the model created as a template from which an automated system will be developed, it is clear that such a system will have the characteristics of a distributed system.

When investigating the actions involved in the process, it is evident that the process itself is distributed in nature. Using software engineering terms, there are three categories of actors involved in the process; the Editor/ Editors, the Submitter, the Reviewers. Without one of these actors, the review process would be incomplete. Considering that each of these actors, in an electronic setting, (*i.e.* through a networked system) could communicate across networked computers though a series of messages being passed, the architecture of such a distributed system quickly begins to evolve.

All communication between Editors, Reviewers and Submitters can, and one could argue for the sake of efficiency, should occur across a network, *e.g.* the Internet. Again using software engineering terms to reflect the process, there are only two types of messages that need to be passed between these actors; Reports/ forms and the submissions themselves. Furthermore, there are only two truly viable options regarding the medium through which this communication occurs, through email or through some variety of 'server side scripting' on the Internet.

## 4 DESIGN

The following chapter outlines the major design decisions that contributed to the final implementation of the framework.

### 4.1 Deployment Diagram

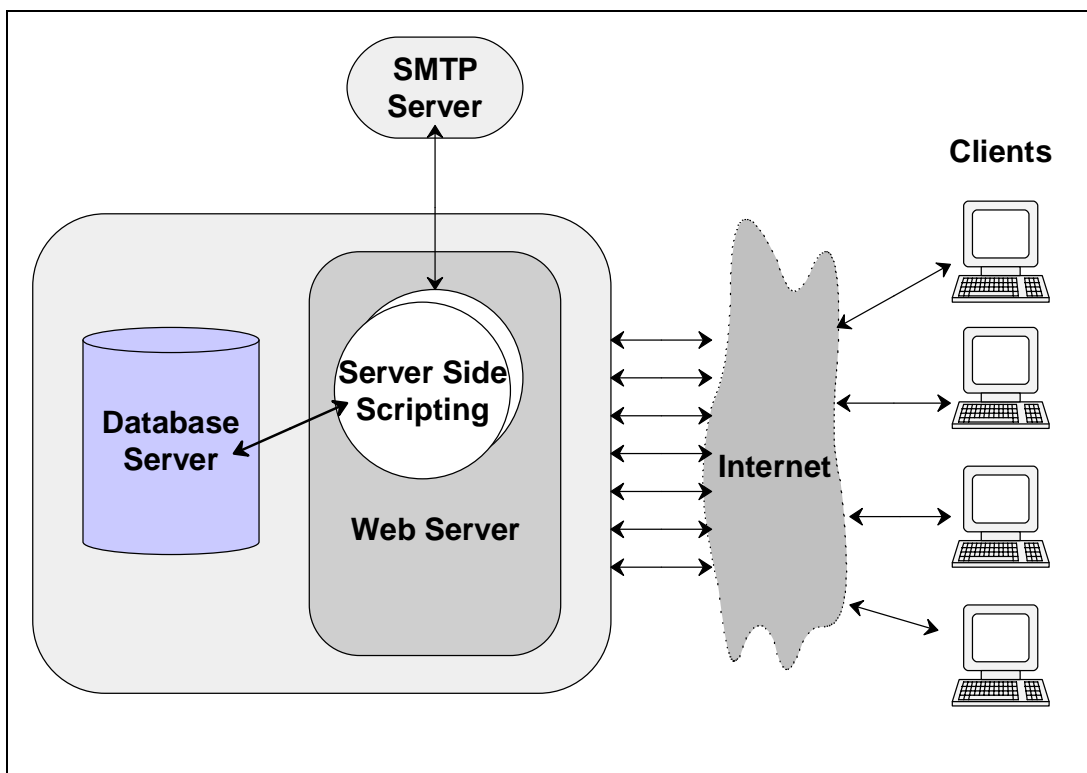


Figure 4.1 Framework Deployment Diagram

The above diagram describes a physical overview of the design of the system. The system works off a standard client server architecture, where clients/ users of the system can interact with persistent data within the server, using the web as a transport layer for requests, and responses. At the core of the system is a database server that holds all pertinent information relating to the submission, policies and management of the journal that it represents. A web server is then used to serve communication between the users and the system, with server side scripts, acting as a dynamic mediator with the web server. Server side scripting is also used to dynamically and automatically create emails based on

information within the database server and information provided by the users, accessing the SMTP server.

All of the technologies in the design are freely available and platform independent. No assumptions need to be made regarding the actual location of the servers. However each server technology should have direct access to each other.

Communication between the system and the clients can occur using any browser technology over the Internet, using HTTP as a transport.

## 4.2 Data Structure Design

Data for the system is stored in two separate forms. Firstly, there is a database that is populated with information that the system uses to both define the peer review process that a journal subscribes to, and to manage and maintain the paper flow of submissions and reports/reviews and the communication between the users of the system. Secondly, there is a hierarchical directory file system that holds all submissions and reports.

### 4.2.1 Database Design

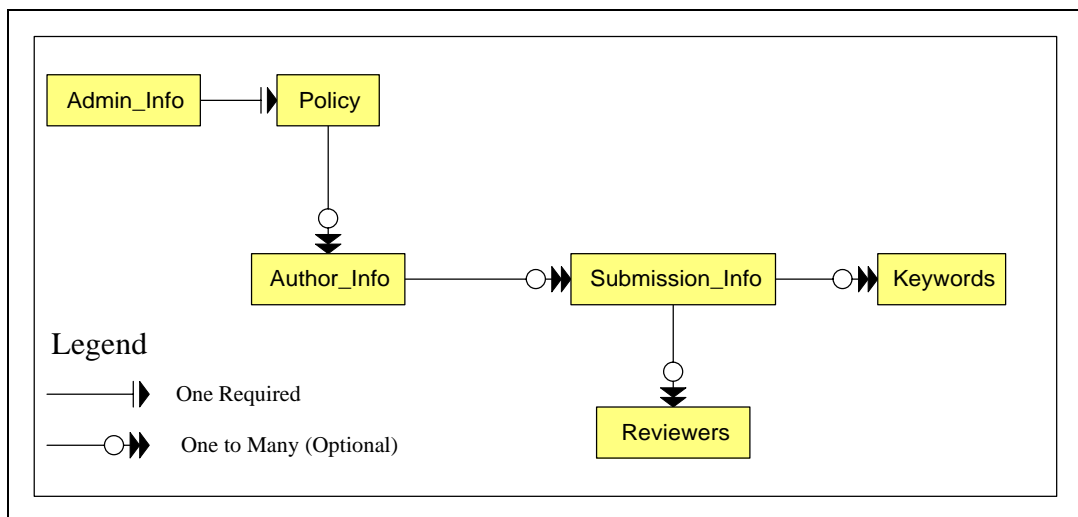


Figure 4.2. Database Design

The previous diagram shows the database design that the system uses. There are five different tables that interact to dictate both the information that the system can store and the policies that define the organisation of a journal's peer review process.

Following are descriptions of the structure of the tables and the semantic meaning of the information that they hold.

### ***Admin\_Info Table***

Field	Type	Null	Key	Default
admin_user	text	YES		(NULL)
password	text	YES		(NULL)
email	text			

The *Admin\_Info* Table holds information that defines the account of the administrator/ editor in chief of the system. The *admin\_user* field holds the username that the administrator uses to log into the system, while the *password* field holds the password that the administrator uses to verify their identity. The *email* field holds the email address of the administrator, which is used primarily in the automation of emails from the system. It is accessed and used as both the 'Reply-to' address, and the 'Sender' address for each email from the system.

### ***Policy Table***

Field	Type	Null	Key	Default
Anon_Review	integer	YES		(NULL)
Internal_Review	integer	YES		(NULL)
Anon_Report	integer	YES		(NULL)
Resubmission	integer	YES		(NULL)
Copyright_Request	integer	YES		(NULL)
Receipt_of_Sub	text	YES		(NULL)
Request_Reviewer	text	YES		(NULL)
Rejection	text	YES		(NULL)
Nice_Rejection	text	YES		(NULL)
Resub_Request	text	YES		(NULL)
Acception	text	YES		(NULL)
Copyright_Accept	text	YES		(NULL)

The *Policy* table acts as the control centre for the system. The contents define the control policies, which determine both the flow of submissions through the peer review process and the contentious issues surrounding communication between the journal’s editors, reviewers and authors. *Anon\_Review* is a Boolean flag that dictates whether submissions are reviewed anonymously or not. In a similar fashion the values of *Internal\_Review*, *Anon\_Report* and *Copyright\_Request*, determine whether the system uses internal reviewing, sends anonymous reviews to authors, and whether the journal asks for a transfer of copyright once a submission is accepted for publication. The value of *Resubmission* controls how many times, if any an author can resubmit their work. Finally, the other fields in the table hold the default text for the possible automated emails of the system. The administrator can change all of these values dynamically in the system through an interface.

#### ***Author\_Info* Table**

Field	Type	Null	Key	Default
Author_ID	<i>character(30)</i>		PRI	
Last_name	<i>text</i>			
first_name	<i>text</i>			
email	<i>text</i>			
password	<i>character(8)</i>			
bio	<i>text</i>			
affiliation	<i>text</i>			
web	<i>text</i>			

This table holds the information on all of the registered authors of the system. The *Author\_ID* field is a unique identifier for each author that is created when an author registers with the system. The rest of the fields are self explanatory, they hold pertinent details on each author, *i.e.* the author’s first and last names, the email address of the author, the password that the author uses to gain access to their account, a biography that gives the editors a profile of the author. The *affiliation* field is used to record any affiliations that the authors might have with any academic organisations, while the *web* address is optional field in case the author might want to reference their homepage.

Usually the authors themselves will edit this table, through a simple interface in the system. They can update their profile in their account at any stage, in case any of the

originally registered fields become out of date, *e.g.* their primary email address. This table is used as a reference point for any submissions that the author may have in the system.

**Submissions Table**

Field	Type	Null	Key	Default
Sub_ID	<i>character(40)</i>		PRI	
title	<i>text</i>			
R_status	<i>character(12)</i>			
last_name	<i>text</i>			
first_name	<i>text</i>			
sub_date	<i>character(8)</i>	YES		(NULL)
location	<i>text</i>	YES		(NULL)
reviewers	<i>text</i>	YES		(NULL)
keywords	<i>text</i>	YES		(NULL)
email	<i>text</i>			
reports	<i>text</i>	YES		(NULL)
anon_reports	<i>text</i>	YES		(NULL)
reports_received	<i>text</i>	YES		(NULL)
reports_rejected	<i>text</i>	YES		(NULL)
Anon_location	<i>text</i>	YES		(NULL)
resub_num	<i>integer(11)</i>	YES		0
resub_date	<i>character(8)</i>	YES		0

The Submissions table holds all relevant information used to control, reference and describe all submissions to the system. The *Sub\_ID* is a unique identifier created upon the any submission by an author of the system. The *title* is simply a holder for the title of the submission. *R\_status* defines the stage of the review process (*e.g.* rejected), which the submission is currently undergoing. This field can have six different values, as defined by the system; new, under review, reviewed, editing, accepted and rejected. The *last\_name*, *first\_name* and *email* fields relate to the name and email of the submitter of the submission, and the *sub\_date* field records the date when the initial submission took place. The *resub\_num* field keeps track of how many times, if any the submission was resubmitted. The *resub\_date* field holds the latest resubmission date, if any for the submission. The *reviewers*, *reports\_received* and *reports\_rejected* hold the names of reviewers requested to review the submission, the reviewers that returned reports/ reviews of the submission and the reviewers that rejected the request to review the submission. The *location* and *anon\_location* hold the physical locations in the hierarchical directory file system, of the submission, and when appropriate, the location of the anonymous version of the submission. Accordingly, the fields *report* and *anon\_report* hold a list of a possible multiple of locations of both reports and

anonymous reports written on the submission. Finally, the *keywords* field holds a list of keywords that the author chose to define their submission. This field is primarily used to aid the editor in choosing which reviewers would be the most appropriate to review the submission.

### **Reviewers Table**

Field	Type	Null	Key	Default
first_name	text			
last_name	text			
email	character(45)			
keywords	text			
Reviewer_ID	integer(20)		PRI	
password	text			

This table holds information for the reviewers' accounts. The administrator creates new accounts for the reviewers associated with the journal. While the reviewers themselves can access and edit their account profile. The *Reviewer\_ID* field is a unique identifier for each reviewer in the system, and is automatically created when an account is created for a reviewer. The *first\_name* and *last\_name* fields are used to identify the reviewers in system and are used primarily to keep track of reviewers associated with submissions. The reviewer uses the *email* and *password* fields to log onto the system and manage their account. Also, the *email* field is used for automatically emailing requests to review submitted material.

### **Keyword Table**

Field	Type	Null	Key	Default
keyword_ID	character(13)		PRI	
keyword	text			

This table, while being the smallest table, is a potentially very usable component of the system. It holds a list of keywords, which the administrator can use to define scholarly areas that the journal aims at addressing. Authors can use the created keywords to define what area their work addresses, while the administrator/ editor can use these as a reference to aid in assigning reviewers to submissions. The *keyword\_ID* field is a unique identifier for each *keyword* and is created automatically when an administrator defines a new keyword.



## 4.2.2 File Directory System

There are seven distinct directories for holding files, *i.e.* submissions and reports, in the system. They are named; *Submitted*, *Being\_Reviewed*, *Reviewed*, *Editing*, *Accepted*, *Rejected* and *Reports*. The first six hold the submissions to the system as they proceed through different phases in the review process and are named accordingly. While the *Reports* directory hold all reports submitted by reviewers to the system.

## 4.3 Use Case Realizations

The following UML use case diagrams define the users of the system and encapsulate the available use-cases that they can realize. There are three different users, an administrator, author, and reviewer, with each defining a user sensitive structure within the overall system. The following diagrams act as a foundation for the subsequent architectural design descriptions which will expand on the use cases.

### 4.3.1 Administrator/ Editor Use Cases

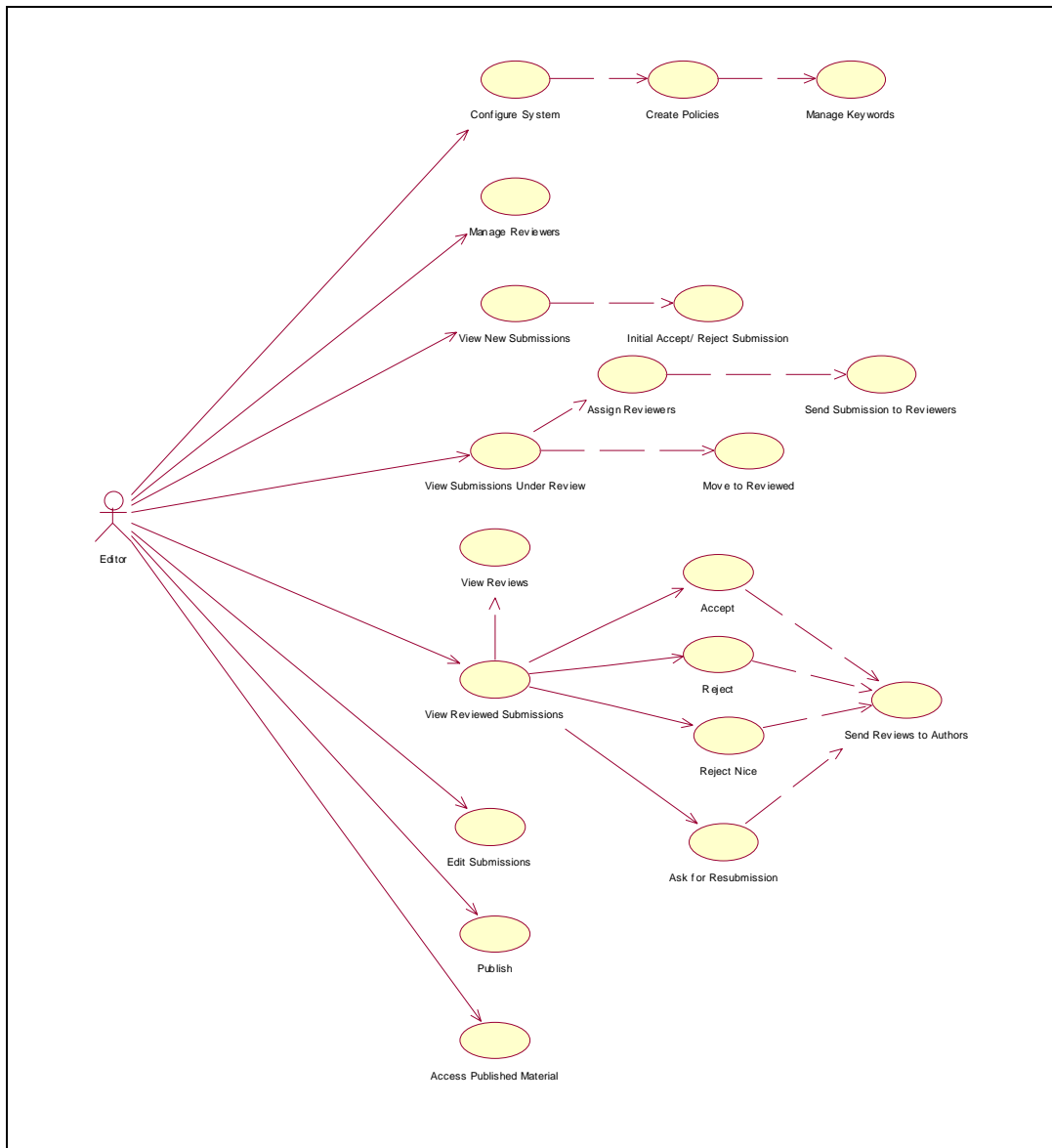


Figure 4.3. Editor Use Cases

### 4.3.2 Author Use Cases

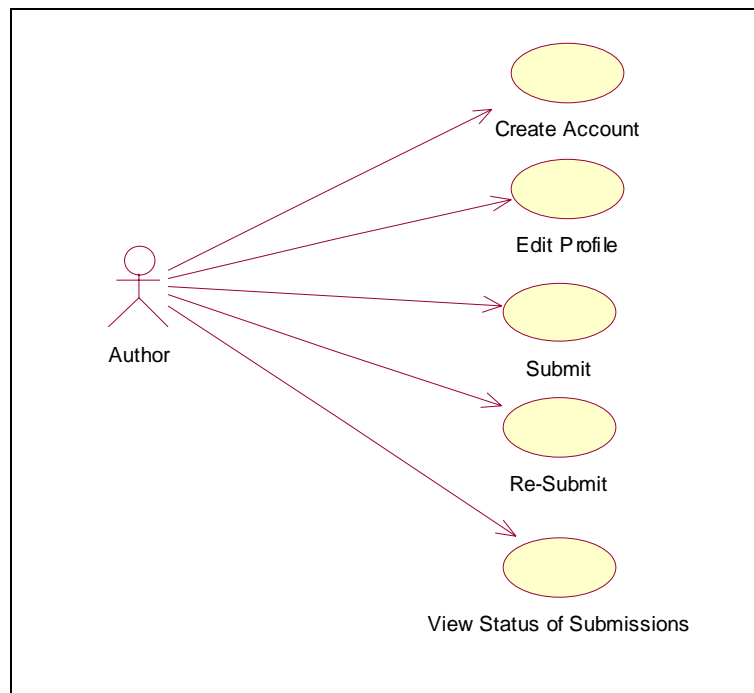


Figure 4.4. Author Use Cases

### 4.3.3 Reviewer Use Cases

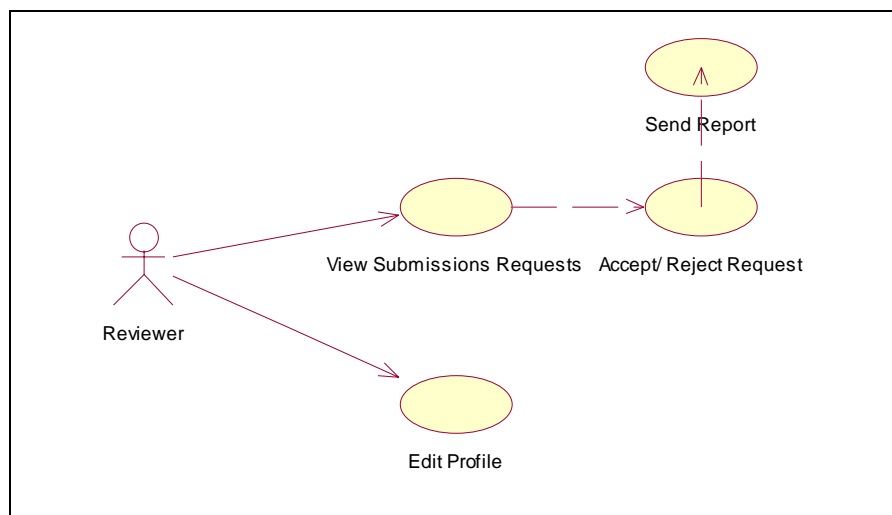


Figure 4.5. Reviewer Use Cases

## **4.4 Architectural Design**

This section will expand on the previous UML Use Case diagrams and provide a walk-through description of the realisation of the use cases and the design components that implement them.

The following UML logical design models define the associations and dependencies between the various components in each user system. The internal components are defined by their names, attributes and operations. Following each model is a written breakdown of each component's attributes, operations, the conditions surrounding each operation and the flow of events through each operation. Select screen shots of the final implementation of the design can be found in the appendices.

## 4.4.1 Administrator/ Editor System

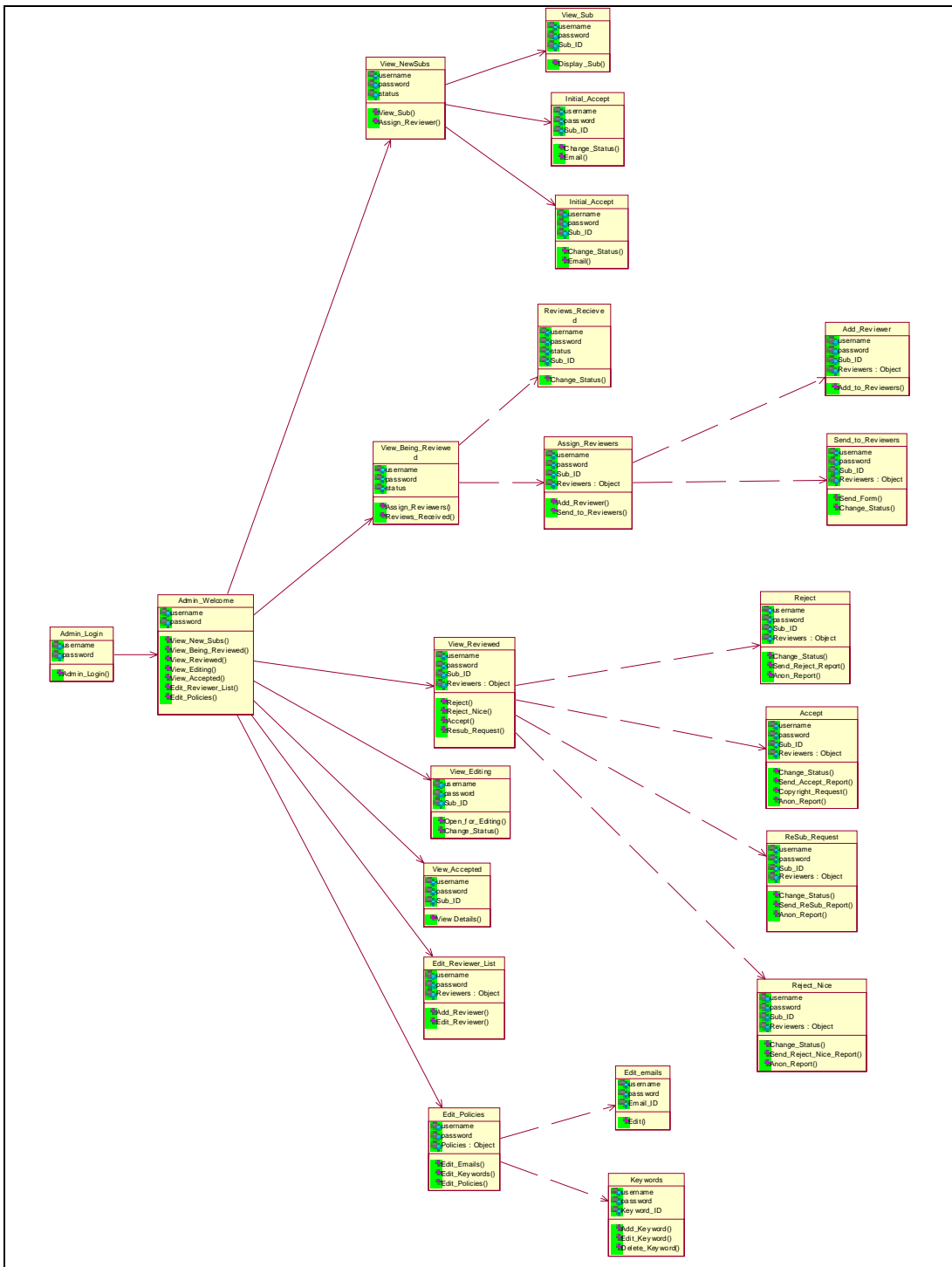


Figure 4.6. Administrator System

**Name:** Administrator Login (*Admin\_Welcome*)

**Type:** Interface

**Description:**

Once the Administrator logs into the system, he/ she is presented with a number of options. These options dictate how the administrator can control the flow of submissions through the peer-review process. The options are logically split up into two areas: viewing the submissions and controlling what stage in the process they reside and secondly, controlling policy related issues in the peer-review process.

**Attributes:**

Administrator Username, Administrator Password.

**Operations:**

Title	View New Submissions ( <i>View_New_Sub</i> s)
Arguments	Administrator Username, Administrator Password.
Pre-Conditions	The Administrator is presently logged into the system.
Post-Conditions	The Administrator is directed to the <i>New Submissions</i> area.
Exceptions	If the Administrator did not login to the system correctly, this option will not be available. The username and password they entered will be shown on the screen and he/ she will be instructed to return to the login page and re-enter the username and password.
Flow of Events	<ol style="list-style-type: none"><li>1. The Administrator presses a button beside the <i>View New Submissions</i> label.</li><li>2. The Administrator is directed to the <i>New Submissions</i> area.</li></ol>

Title	View Submissions Under Review ( <i>View_Being_Reviewed</i> )
Arguments	Administrator Username, Administrator Password.
Pre-Conditions	The Administrator is presently logged into the system.
Post-Conditions	The Administrator is directed to the <i>Submissions Under Review</i> area.

Exceptions	If the Administrator did not login to the system correctly, this option will not be available. The username and password they entered will be shown on the screen and he/ she will be instructed to return to the login page and re-enter the username and password.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator presses a button beside the <i>Submissions Under Review</i> label.</li> <li>2. The Administrator is directed to the <i>Submissions Under Review</i> area</li> </ol>

Title	View Reviewed Submissions ( <i>View_Reviewed</i> )
Arguments	Administrator Username, Administrator Password.
Pre-Conditions	The Administrator is directed to the <i>Reviewed Submissions</i> area.
Post-Conditions	The Administrator is presently logged into the system.
Exceptions	If the Administrator did not login to the system correctly, this option will not be available. The username and password they entered will be shown on the screen and he/ she will be instructed to return to the login page and re-enter the username and password.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator presses a button beside the <i>Reviewed Submissions</i> label.</li> <li>2. The Administrator is directed to the <i>Submissions Reviewed</i> area</li> </ol>

Title	View Submissions Being Edited ( <i>View_Editing</i> )
Arguments	Administrator Username, Administrator Password.
Pre-Conditions	The Administrator is presently logged into the system.
Post-Conditions	The Administrator is directed to the <i>Submissions Being Edited</i> area.
Exceptions	If the Administrator did not login to the system correctly, this option will not be available. The username and password they entered will be shown on the screen and he/ she will be instructed to return to the

	login page and re-enter the username and password.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator presses a button beside the <i>Submissions Being Edited</i> label.</li> <li>2. The Administrator is directed to the <i>Submissions Being Edited</i> area.</li> </ol>

Title	View Accepted Submissions ( <i>View_Accepted</i> )
Arguments	Administrator Username, Administrator Password.
Pre-Conditions	The Administrator is presently logged into the system.
Post-Conditions	The Administrator is directed to the <i>Accepted Submissions</i> area.
Exceptions	If the Administrator did not login to the system correctly, this option will not be available. The username and password they entered will be shown on the screen and he/ she will be instructed to return to the login page and re-enter the username and password.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator presses a button beside the <i>Accepted Submissions</i> label.</li> <li>2. The Administrator is directed to the <i>Accepted Submissions</i> area.</li> </ol>

Title	Edit List of Reviewers ( <i>Edit_Reviewer_List</i> )
Arguments	Administrator Username, Administrator Password
Pre-Conditions	The Administrator is presently logged into the system.
Post-Conditions	The Administrator is directed to the <i>Edit List of Reviewers</i> area.
Exceptions	If the Administrator did not login to the system correctly, this option will not be available. The username and password they entered will be shown on the screen and he/ she will be instructed to return to the login page and re-enter the username and password.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator presses a button beside the <i>Edit List of Reviewers</i> label.</li> </ol>



	2. The Administrator is directed to the <i>Edit List of Reviewers</i> area.
--	---

Title	Control Policy Decisions ( <i>Edit_Policies</i> )
Arguments	Administrator Username, Administrator Password.
Pre-Conditions	The Administrator is presently logged into the system.
Post-Conditions	The Administrator is directed to the <i>Control Policy Decisions</i> area.
Exceptions	If the Administrator did not login to the system correctly, this option will not be available. The username and password they entered will be shown on the screen and he/ she will be instructed to return to the login page and re-enter the username and password.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator presses a button beside the <i>Control Policy Decisions</i> label.</li> <li>2. The Administrator is directed to the <i>Control Policy Decisions</i> area.</li> </ol>

**Name:** View New Submissions Area (*View\_NewSubs*)

**Type:** Interface

**Description:**

This interface provides a listing of all newly submitted works to a journal. The administrator can view the actual documents, and then decide whether or not they would like to initially accept or reject each submission for review. Also, displayed with each submission is a description of the pertinent information relating to the submission, *i.e.* the name of the submitter, the date they submitted the work, any keywords that they might have assigned to the submission.

**Attributes:**

1. Administrator Username.
2. Administrator Password.

3. For Each Submission:

- a. The Submission's Unique Identifier
- b. The location/ locations of the Submission
- c. The First and Last name of the Submitter
- d. The date of submission
- e. Any keywords associated with the submission

**Operations:**

Title	Initial Accept ( <i>Initial_Accept</i> )
Arguments	Administrator Username, Administrator Password, Submission ID
Pre-Conditions	The Administrator is presently logged into the system, and a Submission for which the operation will be performed on is selected.
Post-Conditions	The selected submission is physically moved to the <i>Under Review</i> storage area. The information relating to the submission is updated, <i>i.e.</i> the status of the submission is changed from <i>new</i> to <i>under review</i> .
Exceptions	If the Administrator did not log in properly, they will not be able to have access to this option.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator chooses which Submission they want to accept for review.</li> <li>2. The Administrator presses a button to change the status of the submission.</li> <li>3. The Submission is moved from the <i>New Submissions</i> storage area to the <i>Under Review</i> storage area. <ul style="list-style-type: none"> <li>a. If the system has been set to have anonymous reviews of the submissions, then the anonymous copy of the submission must also be moved to the <i>Under Review</i> storage area.</li> </ul> </li> </ol>

Title	Initial Reject ( <i>Initial_Reject</i> )
Arguments	Administrator Username, Administrator Password, Submission ID
Pre-Conditions	The Administrator is presently logged into the system, and a Submission for which the operation will be performed on is selected.
Post-Conditions	The selected submission is physically moved to the <i>Rejected</i> storage area. The information relating to the submission is updated, <i>i.e.</i> the status of the submission is changed from <i>new</i> to <i>rejected</i> .
Exceptions	If the Administrator did not log in properly, they will not be able to have access to this option.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator chooses which Submission they want to reject.</li> <li>2. The Administrator presses a button to change the status of the submission.</li> <li>3. The Submission is moved from the <i>New Submissions</i> storage area to the <i>Rejected</i> storage area.</li> <li>4. An email is sent to the author, telling them that their submission has been rejected. <ol style="list-style-type: none"> <li>a. If the system has been set to have anonymous reviews of the submissions, then the anonymous copy of the submission must also be moved to the <i>Rejected</i> storage area.</li> </ol> </li> </ol>

**Name:** View Submissions Under Review (*View\_Being\_Reviewed*)

**Type:** Interface

**Description:**

This interface provides a listing of all submissions to a journal that are currently under review. The administrator can view the actual documents, and then decide what reviewers, if any, they would like to have review each submission. Also, displayed with each submission is a description of the pertinent information relating to the submission, *i.e.* the name of the submitter, the date they submitted the work, any keywords that they might have

assigned to the submission, the names of any reviewers already asked to review the submission, the names of any reviewers that may have rejected a request to review a particular submission, the names of the reviewers that returned reports/ reviews on a submission. Once, the administrator decides that they have received all reports/ reviews from the reviewers, they can then move the submission onto the next stage of review, *i.e.* mark the submission as *Reviewed*.

**Attributes:**

1. Administrator Username.
2. Administrator Password.
3. For Each Submission:
  - a. The Submission’s Unique Identifier
  - b. The location/ locations of the Submission
  - c. The First name and Last name of the Submitter
  - d. The date of submission
  - e. Any Reviewers that might have been requested to view the submission
  - f. The names of reviewers that might have returned their reports/ reviews
  - g. The reviews/ reports returned

**Operations:**

Title	Assign Reviewers ( <i>Assign_Reviewers</i> )
Arguments	Administrator Username, Administrator Password, Submission ID
Pre-Conditions	The Administrator is presently logged into the system, and a Submission for which the operation will be performed on is selected.
Post-Conditions	The selected submission is assigned reviewers. The record for the chosen submission is updated accordingly. An email is sent to the reviewer requesting that they review the chosen submission, attached to the email is a copy of the submission.

Exceptions	If the Administrator did not log in properly, they will not be able to have access to this option. If the system is set to have anonymous reviewing, the submission that is sent to the reviewer is an anonymous version of the submission.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator chooses which Submission they want to assign a reviewer to.</li> <li>2. The administrator chooses which reviewers would be best suited to review the chosen submission, and assigns them as reviewers.</li> <li>3. A standardised, but editable, email is sent to the chosen reviewers asking them to review the chosen submission. A copy of the submission is attached to the email. <ol style="list-style-type: none"> <li>a. If the system is set to have anonymous reviewing, the copy attached to the email is an anonymous copy.</li> </ol> </li> <li>4. The record for the submission is updated, noting that the selected reviewers have been requested to review the submission.</li> </ol>

Title	Reviews Received ( <i>Reviews_Received</i> )
Arguments	Administrator Username, Administrator Password, Submission ID
Pre-Conditions	The Administrator is presently logged into the system, and a Submission for which the operation will be performed on is selected. All reviews/ reports, if any, have been received by the system for the selected submission.
Post-Conditions	The selected submission is moved to the “Reviewed” storage area. The status of the submission is changed from “under review” to “reviewed.” If the system is set to have anonymous reviews, then the anonymous copy of the submission is also moved.
Exceptions	If the Administrator did not log in properly, they will not be able to have access to this option.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator chooses which Submission they want to change the status of.</li> </ol>

	<ol style="list-style-type: none"> <li>2. The submission is then moved to the <i>Reviewed</i> storage area.</li> <li>3. If there is an anonymous copy of the review, this copy is also moved. <ol style="list-style-type: none"> <li>a. The status of the chosen submission is changed from <i>under review</i> to <i>reviewed</i>.</li> </ol> </li> </ol>
--	--

**Name:** View Reviewed Submissions (*View\_Reviewed*)

**Type:** Interface

**Description:**

This interface provides a listing of all submissions to a journal that have been reviewed. Displayed with each submission is a description of the pertinent information relating to the submission, *i.e.* the name of the submitter, the date they submitted the work, the names of any reviewers that reviewed the submission and the actual reports/ reviews they submitted. The administrator can view the actual documents, the reports (if any) that were submitted, and then make an informed decision as to the final outcome of the submission. The administrator can have up to four options: Accept the submission for publication, reject the submission for publication, reject the submission, but encourage further submissions and finally, ask the author to resubmit their submission taking advice from the reports/ reviews returned.

**Attributes:**

1. Administrator Username.
2. Administrator Password.
3. For Each Submission:
  - a. The Submission's Unique Identifier
  - b. The location of the Submission
  - c. The First name and Last name of the Submitter
  - d. The date of submission
  - e. The names of reviewers that might have returned their reports/ reviews

- f. The reviews/ reports returned

**Operations:**

Title	Accept ( <i>Accept</i> )
Arguments	Administrator Username, Administrator Password, Submission ID
Pre-Conditions	The Administrator is presently logged into the system, and a Submission for which the operation will be performed on is selected.
Post-Conditions	The selected submission is moved to the <i>Editing</i> storage area. The status of the submission is changed to <i>editing</i> . An email is sent to the author informing them that their submission has been accepted for publication, along with a copy of the reports/ reviews.
Exceptions	<p>If the Administrator did not log in properly, they will not be able to have access to this option. If the system has been set to have anonymous reports/ reviews of submissions, then the attached reports/ reviews will be anonymous versions of the reports.</p> <p>If the journal has a policy of asking for copyright permission to publish a submission, after it has been accepted then the acceptance email will also contain a request relating to this.</p>
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator chooses which Submission they wish to accept.</li> <li>2. An email is sent to the author, with the reports attached. <ol style="list-style-type: none"> <li>a. If the system has been set to have anonymous reports/ reviews of submissions, then the attached reports/ reviews will be anonymous versions of the reports.</li> <li>b. If the journal has a policy of asking for copyright permission to publish a submission, after it has been accepted then the email will also contain a request relating to this.</li> </ol> </li> <li>3. The record containing information about the chosen submission is updated, so that the acceptance is noted.</li> </ol>

	<p>4. The submission is moved to the <i>Editing</i> storage area.</p> <p>a. If the journal has a policy of having anonymous reviewing of submissions the anonymous version of the submission also has to be moved to the <i>Editing</i> storage area.</p>
--	---

Title	Reject ( <i>Reject</i> )
Arguments	Administrator Username, Administrator Password, Submission ID
Pre-Conditions	The Administrator is presently logged into the system, and a Submission for which the operation will be performed on is selected.
Post-Conditions	The selected submission is moved to the <i>Rejected</i> storage area. The status of the submission is changed to <i>rejected</i> . An email is sent to the author informing them that their submission has been rejected for publication, along with a copy of the reports/ reviews.
Exceptions	If the Administrator did not log in properly, they will not be able to have access to this option. If the system has been set to have anonymous reports/ reviews of submissions, then the attached reports/ reviews will be anonymous versions of the reports.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator chooses which Submission they wish to reject.</li> <li>2. An email is sent to the author, with the reports attached. <ol style="list-style-type: none"> <li>a. If the system has been set to have anonymous reports/ reviews of submissions, then the attached reports/ reviews will be anonymous versions of the reports.</li> </ol> </li> <li>3. The record containing information about the chosen submission is updated, so that the rejection is noted.</li> <li>4. The submission is moved to the <i>Rejected</i> storage area. <ol style="list-style-type: none"> <li>a. If the journal has a policy of having anonymous reviewing of submissions the anonymous version of the submission also has to be moved to the <i>Rejected</i> storage area.</li> </ol> </li> </ol>



Title	Reject with Encouragement ( <i>Reject_Nice</i> )
Arguments	Administrator Username, Administrator Password, Submission ID
Pre-Conditions	The Administrator is presently logged into the system, and a Submission for which the operation will be performed on is selected.
Post-Conditions	The selected submission is moved to the <i>Rejected</i> storage area. The status of the submission is changed to <i>rejected</i> . An email is sent to the author informing them that their submission has been rejected for publication, along with a copy of the reports/ reviews.
Exceptions	If the Administrator did not log in properly, they will not be able to have access to this option. If the system has been set to have anonymous reports/ reviews of submissions, then the attached reports/ reviews will be anonymous versions of the reports.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator chooses which Submission they wish to reject.</li> <li>2. An email is sent to the author, with the reports attached. The difference between this operation and a straight rejection, is that the standard email is more encouraging. <ol style="list-style-type: none"> <li>a. If the system has been set to have anonymous reports/ reviews of submissions, then the attached reports/ reviews will be anonymous versions of the reports.</li> </ol> </li> <li>3. The record containing information about the chosen submission is updated, so that the rejection is noted.</li> <li>4. The submission is moved to the <i>Rejected</i> storage area. <ol style="list-style-type: none"> <li>a. If the journal has a policy of having anonymous reviewing of submissions the anonymous version of the submission also has to be moved to the <i>Rejected</i> storage area.</li> </ol> </li> </ol>

Title	Resubmission Request ( <i>Resub_Request</i> )
Arguments	Administrator Username, Administrator Password, Submission ID
Pre-Conditions	The Administrator is presently logged into the system, and a Submission for which the operation will be performed on is selected.
Post-Conditions	The selected submission is moved to the <i>Being Reviewed</i> storage area. The status of the submission is changed to “resubmission.” An email is sent to the author asking them to resubmit their work taking the reports returned into consideration, along with a copy of the reports/reviews.
Exceptions	If the Administrator did not log in properly, they will not be able to have access to this option. If the system has been set to have anonymous reports/ reviews of submissions, then the attached reports/reviews will be anonymous versions of the reports.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator chooses which Submission they wish to ask for a resubmission.</li> <li>2. An email is sent to the author, with the reports attached.</li> <li>3. If the system has been set to have anonymous reports/ reviews of submissions, then the attached reports/ reviews will be anonymous versions of the reports.</li> <li>4. The record containing information about the chosen submission is updated, so that the resubmission request is noted.</li> <li>5. The submission is moved to the <i>Being Reviewed</i> storage area.</li> <li>6. If the journal has a policy of having anonymous reviewing of submissions the anonymous version of the submission also has to be moved to the <i>Being Reviewed</i> storage area.</li> </ol>

**Name:** View Submissions Being Edited (*View\_Editing*)

**Type:** Interface

**Description:**

This interface provides a listing of all submitted works in the editing stage. The administrator can access the actual documents, and edit them accordingly. Once edited, the administrator can move them into the *Accepted* storage area.

**Attributes:**

1. Administrator Username.
2. Administrator Password.
3. For Each Submission:
  - a. The Submission's Unique Identifier
  - b. The location/ locations of the Submission
  - c. The First and Last name of the Submitter
  - d. The date of submission
  - e. Any keywords associated with the submission

**Operations:**

Title	Change Status ( <i>Change_Status</i> )
Arguments	Administrator Username, Administrator Password, Submission ID
Pre-Conditions	The Administrator is presently logged into the system, and a Submission for which the operation will be performed on is selected.
Post-Conditions	The selected submission is physically moved to the <i>Accepted</i> storage area. The information relating to the submission is updated, <i>i.e.</i> the status of the submission is changed from <i>editing</i> to <i>accepted</i> .
Exceptions	If the Administrator did not log in properly, they will not be able to have access to this option.

Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator chooses which Submission they want to accept for review.</li> <li>2. The Administrator presses a button to change the status of the submission.</li> <li>3. The Submission is moved from the <i>Editing</i> storage area to the <i>Accepted</i> storage area.</li> <li>4. If the system has been set to have anonymous reviews of the submissions, then the anonymous copy of the submission must also be moved to the <i>Accepted</i> storage area.</li> </ol>
----------------	---

**Name:** View Accepted Submissions (*View\_Accepted*)

**Type:** Interface

**Description:**

This interface provides a listing of all accepted submissions. The administrator can access the actual documents.

**Attributes:**

1. Administrator Username.
2. Administrator Password.
3. For Each Submission:
  - a. The Submission's Unique Identifier
  - b. The location/ locations of the Submission
  - c. The First and Last name of the Submitter
  - d. The date of submission
  - e. Any keywords associated with the submission

**Operations:**

Title	View Details ( <i>View_Details</i> )
Arguments	Administrator Username, Administrator Password, Submission ID

Pre-Conditions	The Administrator is presently logged into the system, and a Submission for which the operation will be performed on is selected.
Post-Conditions	Details ( <i>i.e.</i> Submitter name, Title, Date Submitted, associated keywords) of the selected submission are displayed.
Exceptions	If the Administrator did not log in properly, they will not be able to have access to this option.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator presses a button to view details.</li> <li>2. The details are displayed.</li> </ol>

**Name:** Edit List of Reviewers (*Edit\_Reviewer\_List*)

**Type:** Interface

**Description:**

This interface provides a listing of all reviewers for the journal. The administrator can add a reviewer, or edit the details of a chosen reviewer.

**Attributes:**

1. Administrator Username.
2. Administrator Password.
3. For Each Previously Entered Reviewer:
  - a. The Reviewers Unique Identifier
  - b. The last and first name of the Reviewer
  - c. The email address of the Reviewer
  - d. Any keywords associated with the Reviewer

**Operations:**

Title	Add Reviewer ( <i>Add_Reviewer</i> )
Arguments	Administrator Username, Administrator Password.
Pre-Conditions	The Administrator is presently logged into the system, and the user has pressed the add reviewer button in the <i>Edit Reviewers</i> area.
Post-Conditions	A new Reviewer is added to the system.
Exceptions	If the Administrator did not log in properly, they will not be able to have access to this option.
Flow of Events	<ol style="list-style-type: none"><li>1. The Administrator presses the <i>Add reviewer</i> button.</li><li>2. The Administrator enters in details of a new reviewer.<ol style="list-style-type: none"><li>a. The first and last name of the reviewer</li><li>b. The reviewer's email address</li><li>c. The Administrator will also choose what keywords in the system to associate with the reviewer.</li><li>d. A password that the reviewer can use to log into the system.</li><li>e. The Administrator submits the details, and a new record of a reviewer is created with a unique identifier associated with it.</li><li>f. An email is sent to the reviewer outlining the details associated with his/ her account.</li></ol></li></ol>

Title	Edit Reviewer ( <i>Edit_Reviewer</i> )
Arguments	Administrator Username, Administrator Password.
Pre-Conditions	The Administrator is presently logged into the system, and the user has selected the reviewer that they wish to edit the account of, in the <i>Edit Reviewers</i> area.
Post-Conditions	A Reviewers account is updated.

Exceptions	If the Administrator did not log in properly, they will not be able to have access to this option.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator selects a reviewer the <i>edit reviewer</i> button.</li> <li>2. The Administrator can change any of the following in details of the selected reviewer. <ol style="list-style-type: none"> <li>a. The first and last name of the reviewer</li> <li>b. The reviewers email address</li> <li>c. The Administrator will also choose what keywords in the system to associate with the reviewer.</li> <li>d. A password that the reviewer can use to log into the system.</li> </ol> </li> <li>3. The Administrator submits the details, and the record of a reviewer is updated.</li> <li>4. An email is sent to the reviewer outlining the details associated with his/ her account.</li> </ol>

**Name:** Control Policy Decisions (*Edit\_Policies*)

**Type:** Interface

**Description:**

This interface provides a listing of all policy decisions for the journal. The administrator can change these policies, thus dynamically changing the working of the system. Also in this area, the administrator can control the look of the default automated emails and add, edit and delete keywords from the system.

**Attributes:**

1. Administrator Username.
2. Administrator Password.
3. Policies
4. Emails

## 5. Keywords

### Operations:

Title	Edit Policies ( <i>Edit_Policies</i> )
Arguments	Administrator Username, Administrator Password.
Pre-Conditions	The Administrator is presently logged into the system, and the user has pressed the <i>Control Policies</i> button.
Post-Conditions	The system is changed accordingly.
Exceptions	If the Administrator did not log in properly, they will not be able to have access to this option.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator can view all available policy decisions.</li> <li>2. The Administrator can change any of the following policies by the press of a button. <ol style="list-style-type: none"> <li>a. Whether the system has anonymous reviewing</li> <li>b. Whether the system has Internal reviewing.</li> <li>c. Whether the system sends out anonymous reports to submitters.</li> <li>d. How many times, if any, a submitter can resubmit their submission.</li> <li>e. Whether an email has to be sent asking for copyright transfer, once a submission has been accepted.</li> </ol> </li> <li>3. The changes that the administrator has made dynamically changes the working logic of the system</li> </ol>

Title	Edit Keywords ( <i>Edit_Keywords</i> )
Arguments	Administrator Username, Administrator Password.
Pre-Conditions	The Administrator is presently logged into the system, and the user has pressed the <i>Control Policies</i> button.



Post-Conditions	The default keywords for the system are changed accordingly.
Exceptions	If the Administrator did not log in properly, they will not be able to have access to this option.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator can view all available keywords.</li> <li>2. The Administrator chooses whether they want to add, edit or delete a keyword.</li> <li>3. If they want to add a keyword: <ol style="list-style-type: none"> <li>a. They enter in a new keyword in a text box provided and press the <i>add</i> button</li> </ol> </li> <li>4. If they want to edit a keyword: <ol style="list-style-type: none"> <li>a. They select the keyword and press the <i>edit</i> button.</li> <li>b. They then edit the keyword in the box provided</li> <li>c. Once the submit button is pressed the keyword is updated in the system.</li> </ol> </li> <li>5. If they want to delete a keyword: <ol style="list-style-type: none"> <li>a. They select the keyword and press the <i>delete</i> button.</li> <li>b. The keyword is then deleted from the system.</li> </ol> </li> </ol>

Title	Edit Emails ( <i>Edit_Emails</i> )
Arguments	Administrator Username, Administrator Password.
Pre-Conditions	The Administrator is presently logged into the system, and the user has pressed the <i>Control Policies</i> button.
Post-Conditions	The default automated emails are changed accordingly.
Exceptions	If the Administrator did not log in properly, they will not be able to have access to this option.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator can view all available automated emails.</li> <li>2. The Administrator chooses which of the following emails they</li> </ol>

	<p>wish to edit by the selecting them and pressing the <i>edit</i> button.</p> <ol style="list-style-type: none"><li>a. Receipt of Submission email</li><li>b. Straight Rejection email</li><li>c. Rejection with Encouragement email</li><li>d. Resubmission Request email</li><li>e. Acceptation email</li><li>f. Acceptation and copyright request email</li></ol> <ol style="list-style-type: none"><li>3. The administrator can then edit the chosen email in a text box.</li><li>4. Once submitted the persistent copy of the email is updated.</li></ol>
--	---

## 4.4.2 Author System

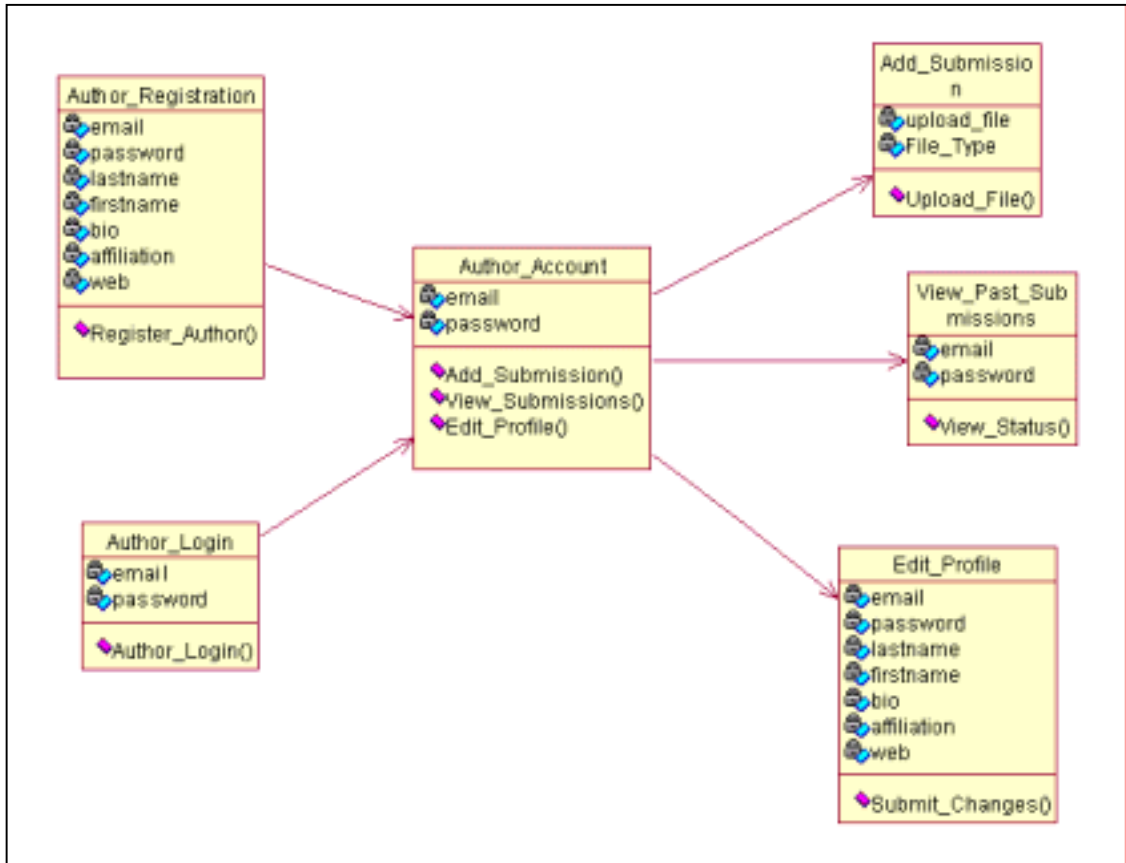


Figure 4.7. Author System

**Name:** Author Registration (*Author\_Registration*)

**Type:** Interface

**Description:**

This interface allows an author to register with the journal as an author. Thus allowing them to submit their work to the journal, and use the author account system.

**Attributes:**

1. First Name
2. Last Name
3. Email Address

4. Password
5. Biography (short description of the author)
6. Affiliation (a listing of any organisations they might be affiliated with, *e.g.* universities)
7. Web Address (their web address, if any)

**Operations:**

Title	Register Author ( <i>Author Registration</i> )
Arguments	Email Address, Author Password, first and last name, biography, affiliation, web address.
Pre-Conditions	The author is accessing the <i>Register Author</i> area.
Post-Conditions	The system records the details that the author submitted. A persistent account is created for the author to use in the future, which they can access using the email address and password they submitted. A unique identifier is created for the author in the system.
Exceptions	If the author fails to enter in the required fields, an error is shown on the screen and they are prompted to fill out the form fully. Otherwise an account will not be created.
Flow of Events	<ol style="list-style-type: none"> <li>1. The author accesses the <i>Author Registration</i> area.</li> <li>2. The author fills out the appropriate fields.</li> <li>3. Once the author submits the information, the system creates a new author account and assigns a unique identifier to the newly registered author.</li> </ol>

**Name:** Add Submission (*Add\_Submission*)

**Type:** Interface

**Description:**

This interface allows a registered author to submit their work to a journal for publication/ review. It will allow the user to upload their submission to the system. The author will be allowed to add a title for the submission, and also assign keywords from a choice of keywords made available, which will help the editors assign appropriate reviewers. The system will mark it as a new submission and create a record relating to the submission containing pertinent information about both the author and the nature of the submission itself.

If the system's policies are set to allow anonymous reviewing of submissions, the author will also be prompted to upload an anonymous version of their submission. Once the submission is uploaded and the record is created, the author will receive an email informing them that the submission was received.

**Attributes:**

1. Author's Email Address.
2. Author's Password.
3. Submission Identifier.
4. Keywords
5. Anonymous Review

**Operations:**

Title	Upload Submission ( <i>Upload_File</i> )
Arguments	Author Email Address, Author Password, Submission ID
Pre-Conditions	The Author is presently logged into the system, and has chosen a file to upload, has entered a title, and has chosen relevant keywords to describe the nature of the submission.
Post-Conditions	The uploaded submission is moved to the <i>New Submissions</i> storage area. A new record is created describing the new submission, <i>i.e.</i>

	<p>the title of the submission, keywords relevant to the submission, the name of the author, the date of submission, the email address of the author, the location of the newly uploaded submission. Once this is completed the author is sent an email, informing them of the system's receipt of the submission.</p>
Exceptions	<p>If the Author did not log in properly, they will not be able to have access to this option. If the system has been set to have anonymous reviewing of submissions, the author will also be prompted to upload an anonymous version of their submission.</p>
Flow of Events	<ol style="list-style-type: none"> <li>1. The author enters the title of the submission and chooses relevant keywords from a list created by the administrator.</li> <li>2. The author then chooses the file he/ she wants to submit for publication and uploads it. <ol style="list-style-type: none"> <li>a. If the system has been set for anonymous reviewing, the author is also prompted to upload an anonymous version of the submission.</li> </ol> </li> <li>3. The submission/s are saved in the <i>New Submissions</i> storage area.</li> <li>4. A record is created in the system, pertaining to the submission.</li> <li>5. The record contains: the title of the submission, keywords relevant to the submission, the status of the submission (<i>i.e.</i> newly submitted), the name of the author, the date of submission, the email address of the author, the location of the newly uploaded submission. <ol style="list-style-type: none"> <li>a. If the system has been set for anonymous reviewing, the location of the anonymous version is also written into the record.</li> <li>b. An email is sent to the author, informing them of the systems receipt of the submission.</li> </ol> </li> </ol>

**Name:** View Past Submissions (*View\_Past\_Submissions*)

**Type:** Interface

**Description:**

This interface allows a registered author to view the status of their previous submissions. Thus, they can keep track of the stage in the editorial process that their submission is currently engaged in.

**Attributes:**

1. Author's Email Address.
2. Author's Password.
3. For each previous submission
  - a. Submission Identifier.
  - b. Associated Keywords
  - c. Status/ editorial stage, of the submission
  - d. Date Submitted

**Operations:**

Title	View Status ( <i>View_Status</i> )
Arguments	Author Email Address, Author Password, Submission IDs
Pre-Conditions	The Author is presently logged into the system, and has to view the status of their submissions.
Post-Conditions	Information on all of their previous submissions is displayed on the screen.
Exceptions	If the author has no previous submissions, there will be no submission information to view.
Flow of Events	<ol style="list-style-type: none"> <li>1. The author chooses to view the status of his/ her submissions</li> <li>2. A list of all previous submissions is displayed, along with the           <ol style="list-style-type: none"> <li>a. Title of the submission</li> <li>b. The Status of the submission</li> <li>c. The data the submission was originally submitted</li> </ol> </li> </ol>

	d. The keywords associated with it.
--	-------------------------------------

**Name:** Edit Author Profile (*Edit\_Profile*)

**Type:** Interface

**Description:**

This interface allows an author to edit their account profile.

**Attributes:**

1. First Name
2. Last Name
3. Email Address
4. Password
5. Biography (short description of the author)
6. Affiliation (a listing of any organisations they might be affiliated with, *e.g.* universities)
7. Web Address (their web address, if any)

**Operations:**

Title	Edit Profile ( <i>Edit_Profile</i> )
Arguments	Email Address, Author Password, first and last name, biography, affiliation, web address.
Pre-Conditions	The author is accessing the <i>Edit Profile</i> area.
Post-Conditions	The system records the details that the author edited. The persistent account is updated, using the author's unique identifier as a reference.
Exceptions	If the author fails to enter any of the required fields, an error is shown on the screen and they are prompted to fill out the form fully.



	Otherwise an account will not be edited.
Flow of Events	<ol style="list-style-type: none"><li>1. The author accesses the <i>Edit Profile</i> area.</li><li>2. The author fills out the appropriate fields.</li><li>3. The author submits their changes.</li><li>4. The persistent account is updated.</li></ol>

### 4.4.3 Reviewer System

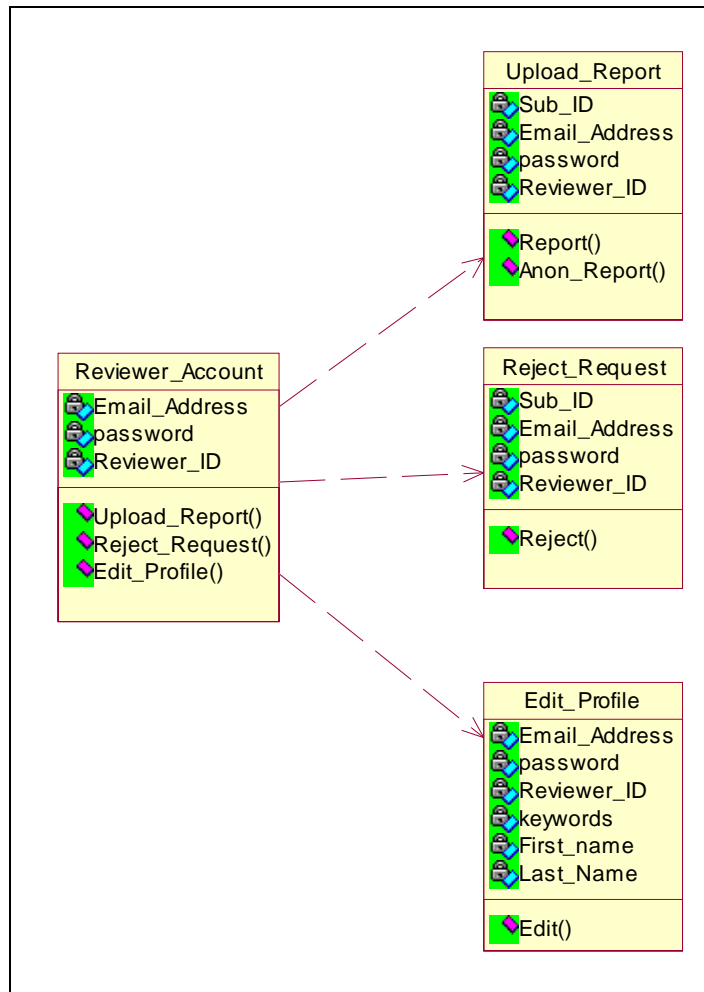


Figure 4.8. Reviewer System

**Name:** Reviewer Account (*Reviewer\_Account*)

**Type:** Interface

**Description:**

Once the Reviewer logs into the system, he/ she can accept or reject requests from the editors to review certain submissions. From this point on the reviewer can submit their reports. Also, the reviewer can change their profile from the login page, *e.g.* if their email address changes, or they wish to change their password.

**Attributes:**

1. Reviewer Email.
2. Reviewer Password.

**Operations:**

Title	Accept Request
Arguments	Reviewer Email, Reviewer Password.
Pre-Conditions	The Reviewer is presently logged into the system.
Post-Conditions	The Reviewer is directed to the <i>Report Upload</i> area.
Exceptions	If the Reviewer did not login to the system correctly, this option will not be available. If the reviewer has not been assigned any new submissions to review, this option will not be available to them.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator presses an <i>Accept Request</i> button beside a description of the submission he/ she was assigned.</li> <li>2. The Administrator is directed to the <i>Report Upload</i> area.</li> </ol>

Title	Reject Request ( <i>Reject_Request</i> )
Arguments	Reviewer Email, Reviewer Password.
Pre-Conditions	The Reviewer is presently logged into the system.
Post-Conditions	The record of the submission is updated to note that the reviewer rejected the request to review the assigned submission.
Exceptions	If the Reviewer did not login to the system correctly, this option will not be available. If the reviewer has not been assigned any new submissions to review, this option will not be available to them.
Flow of Events	<ol style="list-style-type: none"> <li>1. The Administrator presses a <i>Reject Request</i> button beside a description of the submission he/ she was assigned.</li> <li>2. The record for the selected submission is updated, noting that the</li> </ol>

	reviewer rejected the request to review the submission.
--	---

Title	Report Upload
Arguments	Reviewer Email, Reviewer Password.
Pre-Conditions	The Reviewer is presently logged into the system, and has accepted to review a selected submission.
Post-Conditions	The reviewer has uploaded a review/ report on the chosen submission. The record of the submission is updated to note that the reviewer has submitted a report on the submission. Also, the record stores the location of that report.
Exceptions	<p>If the Reviewer did not login to the system correctly, this option will not be available. If the reviewer has not been assigned any new submissions to review, this option will not be available to them.</p> <p>If the system is configured to return anonymous reports on submissions to their authors, the reviewer will also be prompted to upload an anonymous version of their report.</p>
Flow of Events	<ol style="list-style-type: none"> <li>1. The reviewer chooses the file on their computer that holds the review of the submission and uploads it to the system. <ol style="list-style-type: none"> <li>a. If the system is configured to return anonymous reports on submissions, to their authors, the reviewer will also be prompted to upload an anonymous version of their report.</li> </ol> </li> <li>2. The report/s are stored in a <i>Reports</i> storage area.</li> <li>3. Once the system receives this report the record pertaining to the reviewed system is updated to note that the reviewer has submitted a report on the submission.</li> <li>4. The location of the report/s is also recorded.</li> </ol>

Title	Edit Profile ( <i>Edit_Profile</i> )
Arguments	Email Address, Password, first and last name, keywords
Pre-Conditions	The reviewer is accessing the <i>Edit Profile</i> area.
Post-Conditions	The system records the details that the reviewer edited. The persistent account is updated using the reviewer's unique identifier as a reference.
Exceptions	If the reviewer fails to enter any of the required fields, an error is shown on the screen and they are prompted to fill out the form fully. Otherwise an account will not be edited.
Flow of Events	<ol style="list-style-type: none"> <li>1. The reviewer accesses the <i>Edit Profile</i> area.</li> <li>2. The reviewer edits the appropriate fields.</li> <li>3. The reviewer submits their changes.</li> <li>4. The persistent account is updated.</li> </ol>

## 5 IMPLEMENTATION

The following chapter outlines the major concerns that emerged during the implementation of the framework described in the previous chapter. The purpose of this chapter is to extend the previous chapter by describing the actual technologies that were used to realize the design.

### 5.1 Server Technologies

Apache version 1.3.28 was used as the web application server. In conjunction with this PHP 4.3.2 was installed as a module within the Apache Server. The system uses a MySQL server, version 3.23 as its database server. All three servers work in conjunction with each other, with Apache serving PHP scripts that access the MySQL server. The versions named above were chosen due to some minor interoperability issues that demanded such a configuration.

### 5.2 File Types

In designing the system, there was an issue concerning how the system would handle submissions of different file types, *i.e.* uploading and storing a variety of file extensions. This problem was overcome using a combination of both a database storage system, and a simple directory file system.

Once a submission is uploaded, the system takes note of its file extension. A new file name is dynamically created using the submission's unique identifier, which maintains the file extension, *e.g.* if an author submits the file *submission.pdf* the system will create a uniquely identifying name *sub122345.pdf*. This file is then stored in the *Submitted* directory, and a field within the record for that submission is used to hold a pointer/ physical address of the file in the directory.

This process ensures that the file submitted by the author, is identical to the file stored in the system, with the exception of its name. In practice, this ensures that an author

can submit a file of any type extension to the system, as long as the journal has access to an application that can read the submitted file type.

As the file moves through the review process, both its physical location changes, (*e.g.* inside the *Submitted* directory) and the pointer referencing it in the submission's record in the system are updated accordingly.

### **5.3 Policies**

The main control policies outlined in the modelling of the peer review process were all catered for in the final version of the implementation. These policies will dictate the flow of the submissions through the process and dynamically reshape the system to deal with the constraints, accordingly. In summation, the policies that can change from journal to journal depending on the guidelines of the organisation are as follows.

1. Whether the journal has anonymous reviewing of submissions
2. Whether the journal has anonymous reports/ reviews returned to authors
3. How many times, if any, the submitter can resubmit their submission
4. When a transfer of copyright from author to journal is requested
5. Whether the journal has Internal board meeting reviewing, or External reviewing

In order, for the implementation to be a flexible solution for all types of journals the above policy decisions were implemented in the system. In line with the architectural design of the system, the administrator of the system can set these policy decisions through a simple interface with the system.

To illustrate the effect that these policy changes can have on the system, the subsequent diagram will give a workflow model view on the final version of the system in the following configuration.

1. Anonymous reviewing of submissions
2. Anonymous reporting of submissions

3. No resubmissions
4. Transfer of copyright occurs on submission of an article
5. External Reviewing of submitted materials



# Sample Workflow Diagram

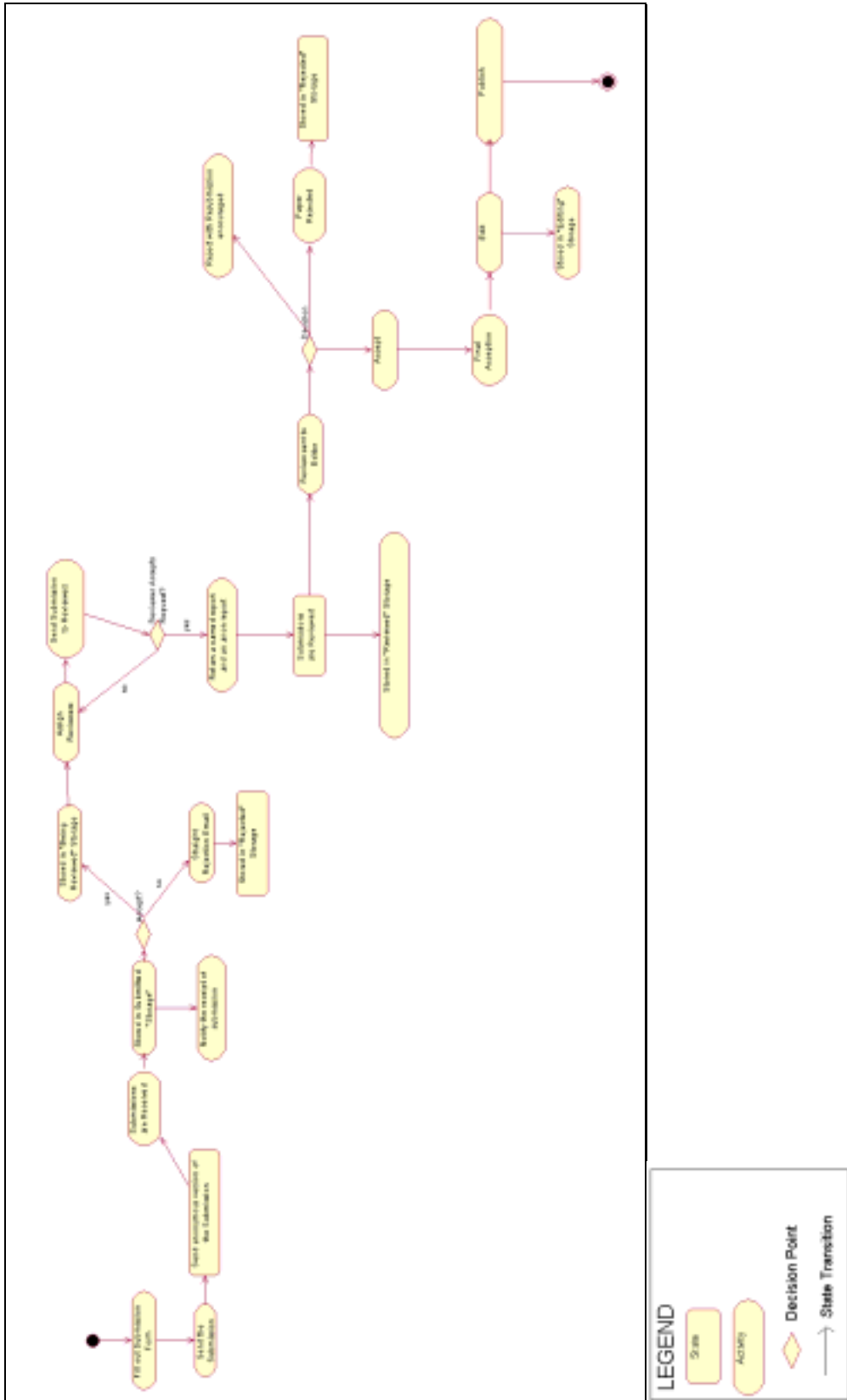


Figure 5.1 Sample Workflow Diagram

The following diagram illustrates the interaction between the users of the system, in the above configuration. The scenario presented, tracks the flow of a submission from the initial submission stage to the acceptance stage.

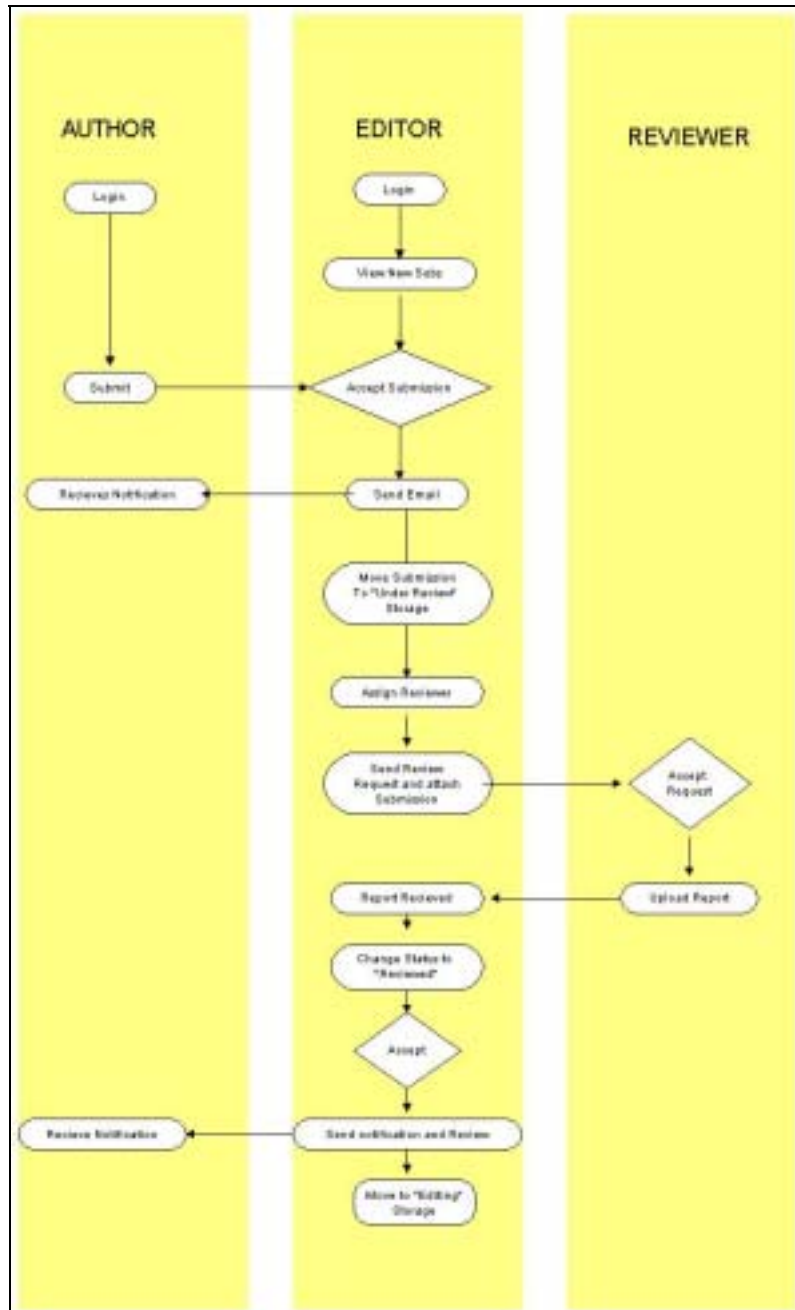


Figure 5.2. Swim-lane Interaction Diagram

## 5.4 Automation

The final version of the implementation addressed the automation concerns noted in chapter 3, in the following ways.

### **Storage:**

In the final implementation the administrator uses a web interface to monitor, manage and direct the submissions through the different outlined storage areas, as noted in the architectural design section.

### **Policies:**

As outlined previously the administrator of the system can define the policies of a journal through a dynamic interface, which affects the system's workflow model in its entirety.

### **Communication:**

The administrator can set up default texts for emails between the system and the authors and the system and the reviewers. Before an email is sent however, the administrator has the option of editing the default text for more subjective communication when needed.

Communication between users that involves the transfer of files, (*e.g.* sending submissions to reviewers, sending reports to authors) is also done dynamically through a simple interface, that attaches the appropriate files (anonymous or otherwise, as defined) to the automatic emails.

The following are the systems default automated emails, with attachments noted.

- a. Receipt of Submission email
- b. Straight Rejection email (reports automatically attached)
- c. Rejection with Encouragement email (reports automatically attached)
- d. Resubmission Request email (reports automatically attached)
- e. Acceptance email (reports automatically attached)
- f. Acceptance and copyright request email (reports automatically attached)

g. Reviewer Request email (submission attached)

## **6 EVALUATION**

The following chapter presents an assessment of the degree to which this project met the goals that were outlined in chapter one. Focus is given to the extent to which the project provides solutions to the concerns that arise in the process of creating, managing and maintaining a scholarly electronic journal.

### **6.1 Modelling**

The workflow model for the peer review process, presented in chapter 3 of this dissertation, provided the project with a defined scope for discussion. Based on both practical experience and focused research, it provided a thoroughly generic reference point from which a journal could devise an individually catered peer review process for their organisational needs. As noted, there had been no previous attempts to model the full workflow process before, and it is hoped that the model will serve as a springboard to further research focusing on improving the efficiency of this integral part of academia.

The model also provided the requirements of the system developed. In order for the system to provide a generic and flexible solution, it had to have the ability to deal with any possible constraints that and/ or permutations of the model.

### **6.2 Framework**

This section addresses the actual implementation of the system. It focuses on how well the system created will be able to adapt to the needs of future users. Again, the aims of the system were to provide a journal with a system that could aid in the creation, management and extension of a journal.

#### **6.2.1 Creating**

Once a journal decides to adopt the framework described in this dissertation, the only development aspects that need to be addressed are configuration issues. Once the server environment is configured to the design specifications outlined in chapter 4 and 5, and the PHP scripts written are stored on the web server, the editor of the journal only needs to make policy decisions and change the default policy changes in the *Control Policies* area. The

system was intentionally designed to be as technologically simple as possible while providing the most amounts of flexibility and functionality possible. It was the intention that the low technological overheads would make the framework accessible to as many journals as possible.

The stages below describe the steps an organisation would need to take to set up the system developed in this project.

1. Install a MySQL server, an Apache web server and PHP.
2. Configure the servers to have permission to access one another.
3. Gain permission to access to a SMTP server.
4. Create a database and tables as described in chapter 4.
5. Insert into the *Admin\_Info* table the username and password desired.
6. Create a directory system as described in chapter 4.
7. Save the source code of the system in the home directory of the web server.
8. Edit the source code, replacing the default username and password used to access the database.
9. Log into the administrator system and configure the policies, keywords and emails as desired by the organisation.

As can be seen above, the steps involved in creating an individually catered peer review system from the system developed during this project, would be a relatively simple exercise for a person with a background in computer science.

## 6.2.2 Managing

The management of the system occurs in the Administrators area. The administrator of the system can manage the flow of submissions through the editorial process, through simple web interfaces. Apart from the editing of emails, there is no typing involved in the

management of the system, as it was developed with the intention of having a simple and intuitive graphical interface.

The functionality of the system depends on the policies configured by the administrator of the system. Therefore, it is advised that the administrator thinks of the repercussions that any policy decisions may have on both the functionality of the system and its implications on its users and the organisation of the journal itself. However, through careful configuration, the system can adapt to the demands of any peer review process. In short, the system can manage the paper-flow of the editing process, the communication between a journal and its reviewers and submitters, and any policy decisions that a journal may subscribe to.

### 6.2.3 Maintaining

As a journal grows and the number of people involved in the production of a journal increases, it is goal of this project to be able to aid the editors of a journal to maintain, a persistent record of both authors, reviewers and articles submitted to the journal, regardless of any organisational changes that may occur. As a journal extends the complexity of its review process, the system can accordingly adapt to any new demands. This was also made possible through the modelling of a generic process. Therefore, to illustrate the implications of the flexibility of the system, the researcher will use the example of an emerging start-up journal.

Hypothetically speaking, a start-up journal may involve a handful of editors that review the submissions themselves, have no external reviewers, and are unconcerned about the implications of anonymous reviewing. However, in time to come the organisation may expand to include external reviewers in the review process and decide to implement a process of anonymous reviewing and reporting. As the system is designed to be inherently flexible, the journal described above could use the system throughout its expansion, and as the organisation expands, the system can also be configured to expand accordingly.

## 6.3 Adoptability

In adopting the system, a journal would require little implementation overhead in creating a specifically catered solution as described above. However, one must note that there is a degree of technical knowledge that is needed in order to adopt the framework.

In order for a journal to create an individual version of the system described in this dissertation, the implementer would need to have basic knowledge in the three following areas:

1. Installation and configuration of a MySQL server, Apache Server and PHP.
2. Simple database creation.
3. Basic PHP knowledge.

Considering that most academic journals are administered within an academic situation, *i.e.* a university, one could assume that where needed, a journal could ask an IT professional in the university to help them set up the system.

## **6.4 Cross Platform Issues**

The system was designed to be as Operating System independent as possible. All of the technologies used in the framework, have versions for most platforms. All files are stored on the web server, thus eliminating any platform dependent directory access issues in the source code. Also, all directory names are valid for most common operating systems.



## **7 CONCLUSIONS**

### **7.1 Goals Achieved**

All goals as described in the introduction and scope of the dissertation have been met with relative success. Previous to this research, there had been no attempt at modelling the academic peer review process in a generic fashion. Furthermore, there was lack of a single point of reference from which journals could explore the possibilities of implementing an electronic journal publication. The research presented in this dissertation will hopefully serve as a foundation from which the marriage of technology and scholarly publications can enhance both the communication and the dissemination of knowledge. It is also hoped that journals will utilise this dissertation and the system developed through this project, to improve the organisation and management of their publication process; the aim being to make their processes more efficient, and lower their publication costs.

### **7.2 Limitations of Research**

The time constraints surrounding the completion of this dissertation, unfortunately led to an unavoidably narrow focus of research. The scope of the project was tapered such that realistic goals could be achieved. It must be noted that if the scope had included the areas of presenting and managing archives of journal material, it would have provided the research with more complete findings. Time constraints also stifled the development of a means to lessen the technological knowledge required to develop a journal specific implementation of the system described.

Moreover, let it be noted that there was a lack of qualitative research in the specification gathering for the implementation of the framework. The research involved in the dissertation was rooted solely in empirical investigation and third party sources.

In addition, there was no pilot study undertaken to investigate the usability of the system developed. Consequently, again due to time constraints there was no acceptance testing organized to examine the impact of the system on the targeted community.

### **7.3 Recommendations for Future Research**

Initially, this researcher would recommend that the system developed be extended to include an automated installation program, for various platforms. With this in place further testing of the system could take place to improve upon the developments achieved.

Another possible extension of the project would be to extend the system to automatically create some form of metadata on submissions that are accepted for publication. The extension could create this metadata and tie the system into the OAI-PMH framework.

In general, it is hoped that the peer review model created could aid the further development of tools to automate the integration of journals with such initiatives as the Semantic Web, the Open Access Initiative and the Open Archives Initiatives.

In conclusion, the aims of the project were to develop a framework for academic electronic journals, and create a system that would aid in the creation, management and extension of any journal. Inherently, it is hoped that this framework can be further built upon.

## 8 APPENDICES

### 8.1 Appendix A: Administrator/ Editor System



Figure 8.1 Administrator Welcome

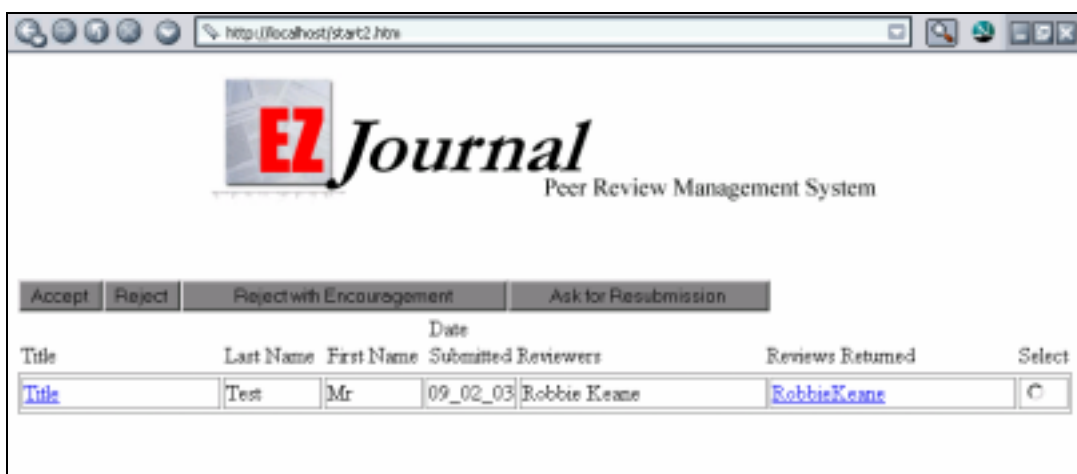


Figure 8.2 Reviews Received Area



Figure 8.3. Edit Reviewers

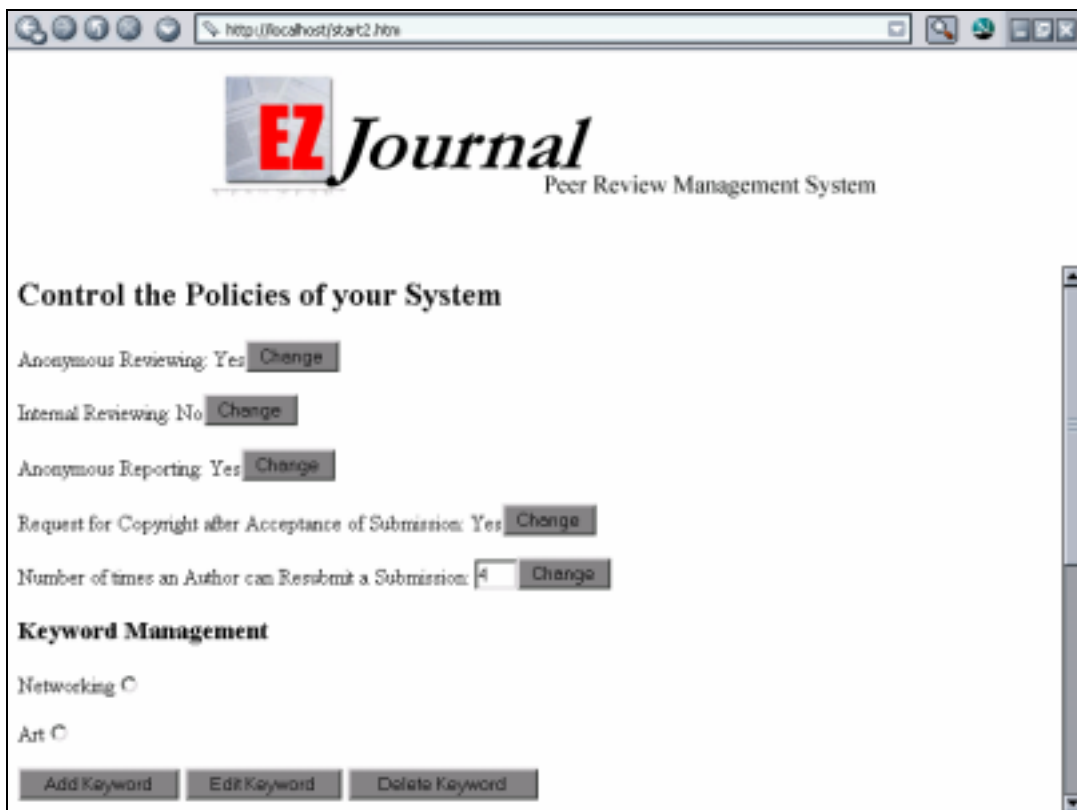


Figure 8.4 Control Policies

## 8.2 Appendix B: Author System

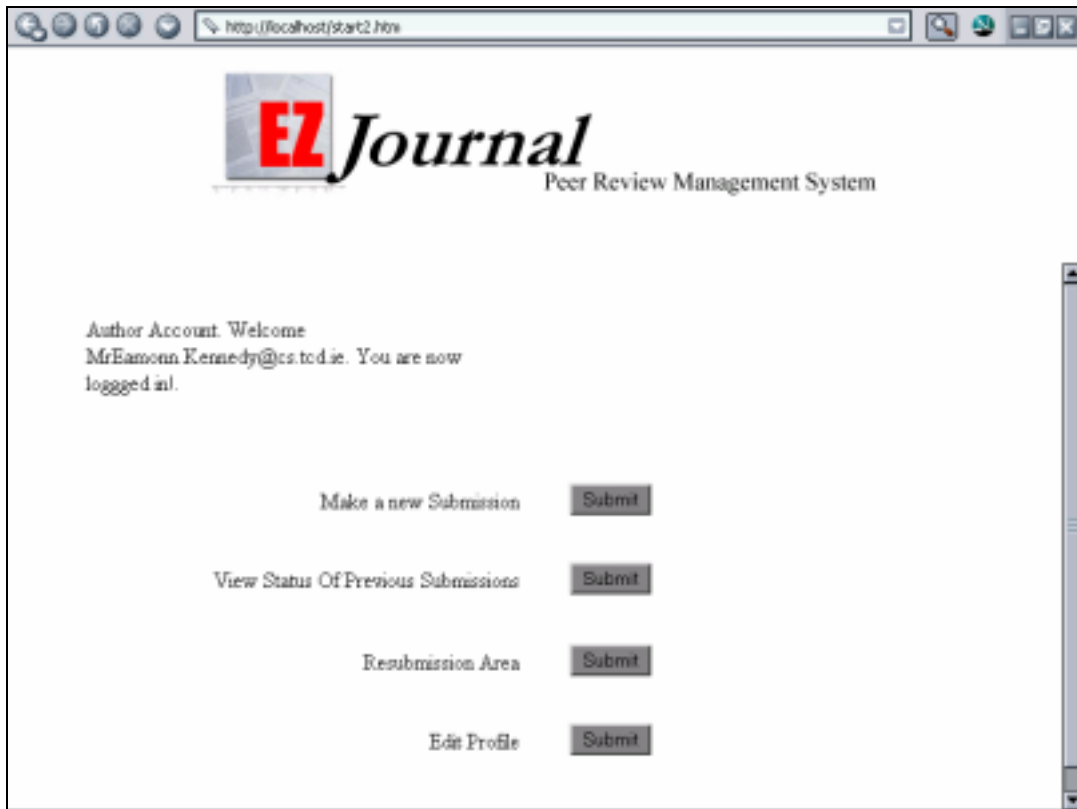


Figure 8.5. Author Account

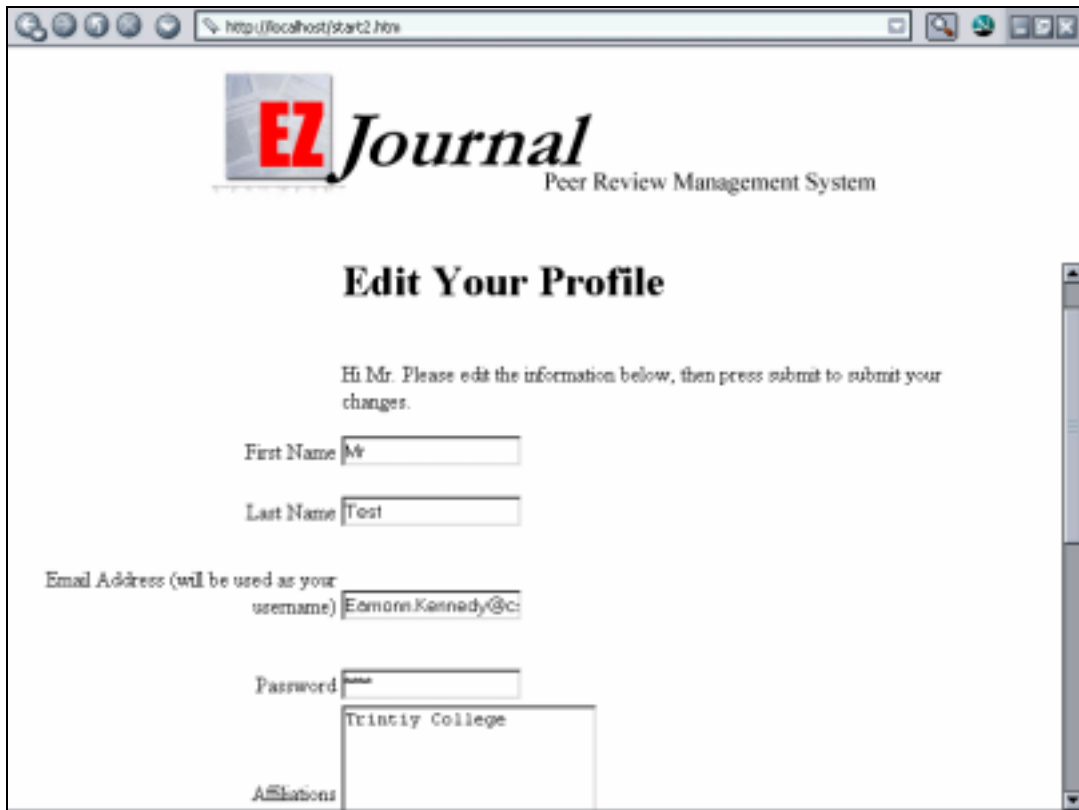


Figure 8.6. Edit Profile



Figure 8.7 View Past Submissions

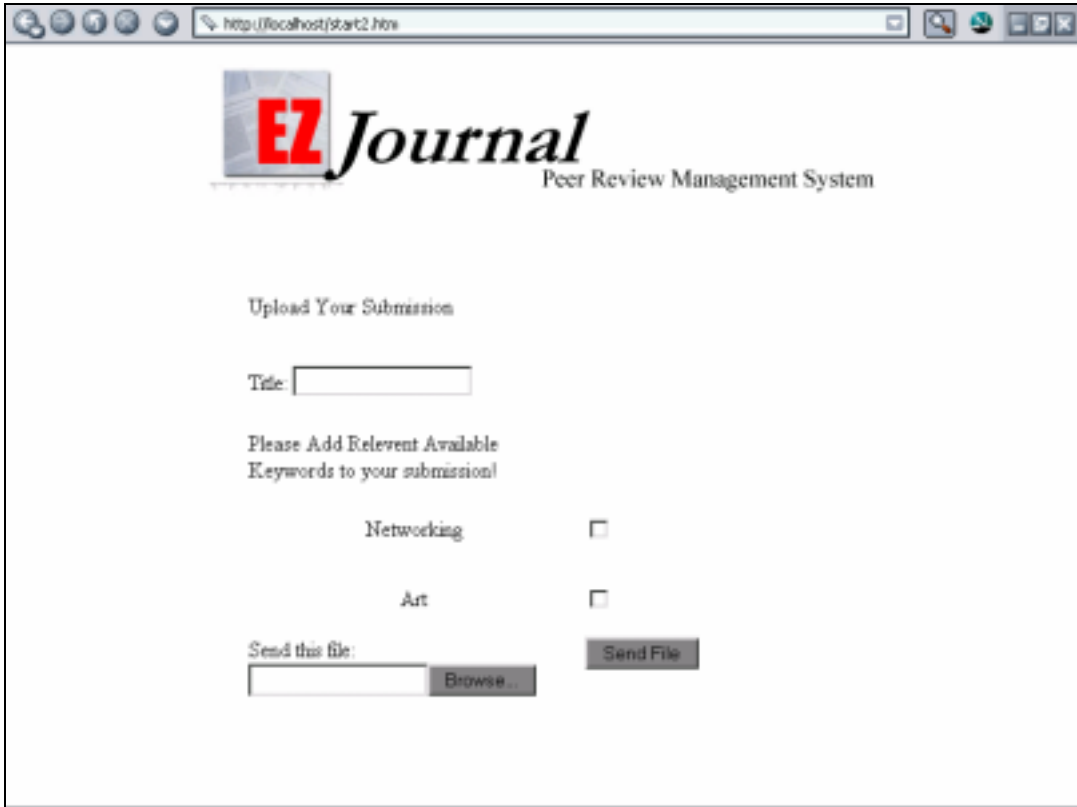


Figure 8.8 Upload Submission

### 8.3 Appendix C: Reviewer System

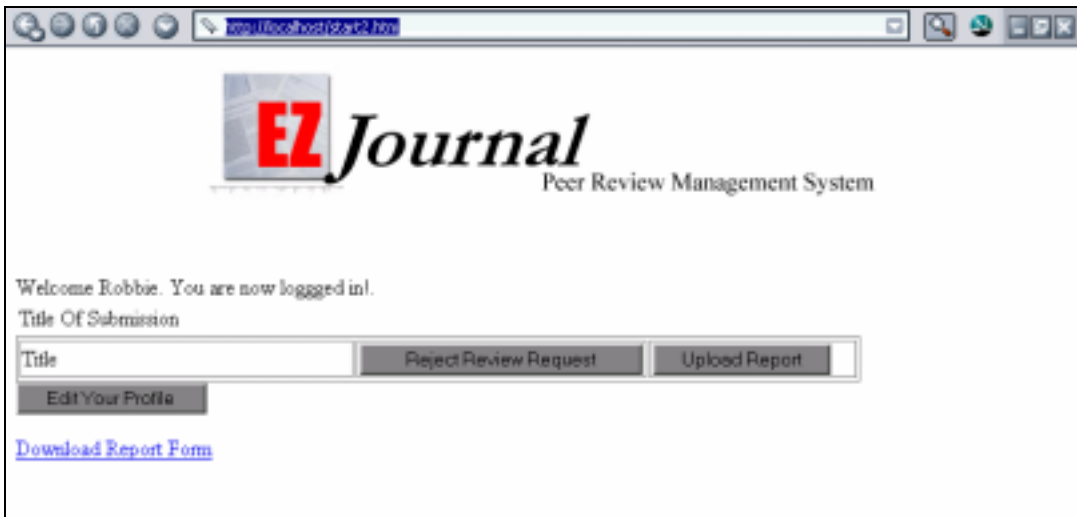


Figure 8.9. Reviewer Account

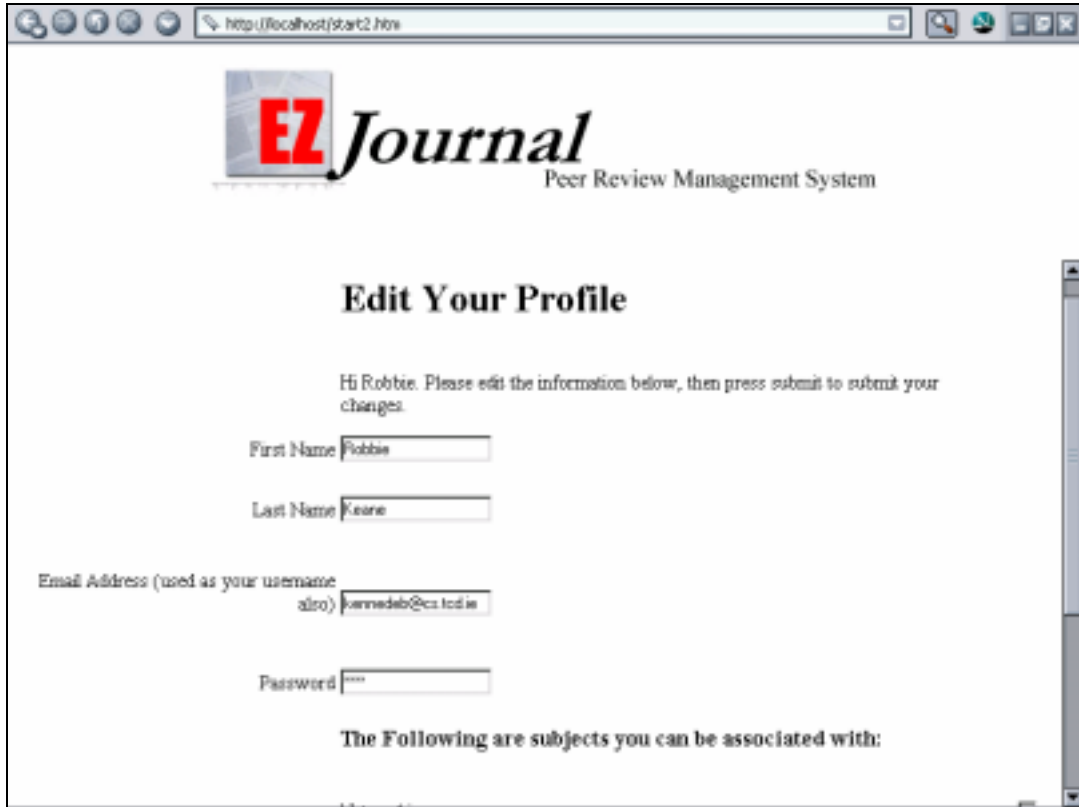


Figure 8.10. Reviewer Edit Profile



## 9 BIBLIOGRAPHY:

1. "Soros Foundations 1999 Annual Report." Open Society Institute. Retrieved 25<sup>th</sup> July, 2003 from [http://www.soros.org/annual/index\\_result.html](http://www.soros.org/annual/index_result.html)
2. "Open Society Institute and Soros Foundations Network." Retrieved 25<sup>th</sup> July, 2003 from <http://www.soros.org/>
3. The Budapest Open Access Initiative Homepage. Retrieved 25<sup>th</sup> July, 2003 from <http://www.soros.org/openaccess/index.shtml>
4. "The Budapest Open Access Initiative." (2002, February 14<sup>th</sup>). Retrieved 25<sup>th</sup> July, 2003 from <http://www.soros.org/openaccess/read.shtml>
5. The Directory of Open Access Journals Home page. Retrieved 25<sup>th</sup> July, 2003 from <http://www.doaj.org/>
6. The Directory of Open Access Journals "About" page. Retrieved 25<sup>th</sup> July, 2003 from <http://www.doaj.org/articles/about/>
7. The Directory of Open Access Journals "FAQ" page. Retrieved 25<sup>th</sup> July, 2003 from <http://www.doaj.org/articles/questions/>
8. The Santa Fe Convention for the Open Archives Initiative Home page (2000, February 15<sup>th</sup>) Retrieved 25<sup>th</sup> July, 2003 from [http://www.openarchives.org/meetings/SantaFe1999/sfc\\_entry.htm](http://www.openarchives.org/meetings/SantaFe1999/sfc_entry.htm)

9. "The Santa Fe Convention for the Open Archives Initiative Core Document" (2000, February 15<sup>th</sup>) Retrieved 25<sup>th</sup> July, 2003 from  
<http://www.openarchives.org/sfc/sfc.htm>
10. Greenstein, D., Lynch, C. (2000, August 15<sup>th</sup>) "The Open Archives Initiative Appoints Steering Committee." Retrieved 25<sup>th</sup> July, 2003 from  
<http://www.openarchives.org/news/oaiscpres000825.html>
11. Van de Sompel, H., Lagoze, C. (2002) "A Progress Report on the Open Archives Initiative." Retrieved 25<sup>th</sup> July, 2003 from  
<http://www.openarchives.org/documents/ecdl-oai.pdf>
12. "About the Dublin Core Metadata Initiative" Retrieved 25<sup>th</sup> July, 2003 from  
<http://dublincore.org/about/>
13. "The Open Archives Initiative Protocol for Metadata Harvesting" (2003, February 21). Retrieved 25<sup>th</sup> July, 2003 from  
<http://www.openarchives.org/OAI/openarchivesprotocol.html>
14. Berners-Lee, T., Hendler, J., Lassila, O., "The Semantic Web" (2001, May). Scientific American Magazine, May 2001 Issue. Retrieved 25<sup>th</sup> July, 2003 from  
<http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2>
15. Manola, F., Miller, E., "RDF Primer" (2003, January 23<sup>rd</sup>). Retrieved 25<sup>th</sup> July, 2003 from <http://www.w3.org/TR/2003/WD-rdf-primer-20030123/>

16. "Igniting Change in Scholarly Communication: SPARC its Past, Present and Future" (2002). Retrieved 25<sup>th</sup> July, 2003 from [http://www.arl.org/sparc/SPARC\\_Advances.pdf](http://www.arl.org/sparc/SPARC_Advances.pdf)
17. "Create Change Home" (2003). Retrieved 25<sup>th</sup> July, 2003 from <http://www.createchange.org/home.html>
18. Johnson, R. (2000). "Declaring Independence." Retrieved from SPARC website July 10<sup>th</sup>, 2003: [http://www.arl.org/sparc/DI/Declaring\\_Independence.pdf](http://www.arl.org/sparc/DI/Declaring_Independence.pdf).
19. Johnson, R. (April, 2002). "Gaining Independence." Retrieved from SPARC website July 10<sup>th</sup>, 2003: [http://www.arl.org/sparc/GI/SPARC\\_GI\\_MANUAL\\_VERSION1.0.PDF](http://www.arl.org/sparc/GI/SPARC_GI_MANUAL_VERSION1.0.PDF)
20. "SPARC Alternatives." Retrieved 25<sup>th</sup> July, 2003 from <http://www.arl.org/sparc/core/index.asp?page=c1>
21. "Current SPARC Members" (July 30, 2003). Retrieved 25<sup>th</sup> July, 2003 from <http://www.arl.org/sparc/core/index.asp?page=b1>.
22. "The Association of Learned and Professional Society Publishers" (July 13, 2003). Retrieved 25<sup>th</sup> July, 2003 from <http://www.alpsp.org/journal.htm>.
23. "Online Peer Review: Current Options." Learned Publishing, Volume 14, April 2001. <http://gottardo.ingentaselect.com/vl=8247840/cl=20/nw=1/fm=docpdf/rpsv/catchword/alpsp/09531513/v14n2/s12/p151>
24. "The ESPERE Project- Online submission and peer review for journals," (2003, July 2<sup>nd</sup>). Retrieved 25<sup>th</sup> July, 2003 from <http://www.espere.org/>

25. "Scholar One – Manuscript Central." Retrieved 25<sup>th</sup> July, 2003 from [http://www.scholarone.com/products\\_manuscriptcentral.html](http://www.scholarone.com/products_manuscriptcentral.html)
26. "Welcome to Rapid Review." Retrieved 25<sup>th</sup> July, 2003 from <http://www.rapidreview.com/>
27. "XpressTrack: Smartware for Journals." Retrieved 25<sup>th</sup> July, 2003 from <http://www.xpresstrack.com/>
28. "The Espere System Author Submissions," (2003, January 4th). Retrieved 25<sup>th</sup> July, 2003 from [http://www.espere.org/auth\\_sub.htm](http://www.espere.org/auth_sub.htm).
29. "The ESPERE Project- Online submission changes," (2003, January 4th). Retrieved 25<sup>th</sup> July, 2003 from [http://www.espere.org/submit\\_change.htm](http://www.espere.org/submit_change.htm)
30. Stephen, T., Harrison, T.M (2002, August). "Building Systems Responsive to Intellectual Tradition and Scholarly Culture." The Journal of Electronic Publishing, Volume 8, Issue 1. Retrieved from <http://www.press.umich.edu/jep/08-01/stephen.html>.
31. Bishop, A.P (1998, December). "Measuring Access, Use, and Success in Digital Libraries." The Journal of Electronic Publishing, Volume 4, Issue 2. Retrieved from <http://www.press.umich.edu/jep/04-02/bishop.html>.
32. Raney, A.A., Jackson, J.R., Edwards, D.B., Schaffler, K.L., Arrington, J.B., Price, M.R. (2002, April). "The Relationship Between Multimedia Features and Information Retrieval." The Journal of Electronic Publishing, Volume 7, Issue 3. Retrieved from <http://www.press.umich.edu/jep/07-03/raney.html>.

33. ARL Directory of Electronic Journals, Newsletters, and Academic Discussion Lists, (n.d). Retrieved July 10<sup>th</sup>, 2003, from <http://www.arl.org/scomm/edir/archive.html>
34. Denning, P.J., (April 2002). "The ACM Electronic Publishing Plan." Retrieved from ACM website July 10<sup>th</sup>, 2003: [http://www.acm.org/pubs/epub\\_plan.html](http://www.acm.org/pubs/epub_plan.html)
35. "ACM Copyright Policy" (November 1<sup>st</sup>, 2002). Retrieved from ACM website July 10<sup>th</sup>, 2003: [http://www.acm.org/pubs/copyright\\_policy/](http://www.acm.org/pubs/copyright_policy/)
36. "IEEE Transactions, Journals, and Letters: Information for Authors." ( January, 2003). Retrieved from IEEE website July 10<sup>th</sup>, 2003:  
<http://a957.g.akamai.net/f/957/3680/1h/www.ieee.org/organizations/pubs/transaction/s/auinfo03.pdf>
37. "Crossings: ejournal of Art and Technology, *Raison d'Être*." Retrieved from Crossings website July 10<sup>th</sup>, 2003: <http://crossings.tcd.ie/mission.shtml>
38. "Crossings: ejournal of Art and Technology, Contributor's Guide." Retrieved from Crossings website July 10<sup>th</sup>, 2003: <http://crossings.tcd.ie/contrib/>
39. Peters, J., (1996, May)"The Hundred Years War Started Today: An exploration of electronic peer review." The Journal of Electronic Publishing. Retrieved from the JEP website July 10<sup>th</sup>, 2003: <http://www.press.umich.edu/jep/works/PeterHundr.html>
40. Peters, J., "Electronic peer review," keynote paper of the Electronic Peer Review Internet Conference, 1995, [www.mcb.co.uk](http://www.mcb.co.uk).
41. Coulouris, G. (2003). "Distributed Systems, Concepts and Design. (3<sup>rd</sup> Edition)." Addison-Wesley Publications.

