AN INTEGRATED INFORMATICS INFRASTRUCTURE FOR PRE-CLINICAL RESEARCH-IT SUPPORT

by

JACEK SZYMANSKI

Submitted in partial fulfillment of the requirements

For the degree of Doctor of Philosophy

Department of Electrical Engineering and Computer Science

CASE WESTERN RESERVE UNIVERSITY

January, 2008
We hereby approve the dissertation of

Jacek Szymanski

candidate for the Ph.D. degree *

Guo-Qiang Zhang

(signed)

(Chair of the Committee)

Kenneth Loparo

Chris Flask

Michael Branicky

(date) 11/20/07

*We also certify that written approval has been obtained for any proprietary material contained therein.
# Contents

1 Introduction to MIMI .......................... 10
  1.1 Core facilities and research-IT .................. 10
    1.1.1 Types and scope of core facilities ........... 13
    1.1.2 Core facilities on the CWRU campus .......... 15
    1.1.3 Information and data management challenges ... 18
      1.1.3.1 The University of Washington study ........ 19
      1.1.3.2 Deficiencies in existing infrastructures ... 21
      1.1.3.3 An ideal informatics infrastructure for research-IT ... 23
      1.1.3.4 Related work ................................ 24
  1.2 Problem statement ................................ 26
  1.3 Contributions of this thesis .................... 28
    1.3.1 A common informatics infrastructure for cores ... 28
    1.3.2 A set of comprehensive features in a single system ... 29
    1.3.3 Data modeling, system development, and deployment ... 30
  1.4 Thesis overview .................................. 30

2 Data Modeling of a Core Facility’s Information Flow 33
  2.1 Introduction ...................................... 33
  2.2 Methods .......................................... 36
    2.2.1 Gathering preliminary information ............ 36
    2.2.2 Building the initial object model ............ 37
      2.2.2.1 Object identification ...................... 38
      2.2.2.2 Object names, attributes, and operations ... 41
      2.2.2.3 Object associations ....................... 42
    2.2.3 Refining the initial object model ............ 43
      2.2.3.1 Applying inheritance ...................... 44
      2.2.3.2 Applying object combination ............... 45
    2.2.4 Assessing effectiveness ....................... 46
  2.3 Results .......................................... 47
  2.4 Discussion ....................................... 49

3 The Imaging MIMI .................................... 51
  3.1 Introduction ..................................... 51
  3.2 Methods .......................................... 53
    3.2.1 Guiding principles and approach .............. 53
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.2</td>
<td>System overview</td>
<td>56</td>
</tr>
<tr>
<td>3.2.3</td>
<td>Design and implementation</td>
<td>57</td>
</tr>
<tr>
<td>3.2.3.1</td>
<td>User profile model</td>
<td>58</td>
</tr>
<tr>
<td>3.2.3.2</td>
<td>Project model</td>
<td>61</td>
</tr>
<tr>
<td>3.2.3.3</td>
<td>Session model</td>
<td>64</td>
</tr>
<tr>
<td>3.2.3.4</td>
<td>Data-flow</td>
<td>69</td>
</tr>
<tr>
<td>3.3</td>
<td>Results</td>
<td>73</td>
</tr>
<tr>
<td>3.3.1</td>
<td>Usage analysis</td>
<td>73</td>
</tr>
<tr>
<td>3.3.2</td>
<td>Performance analysis</td>
<td>75</td>
</tr>
<tr>
<td>3.3.2.1</td>
<td>Transfer rates</td>
<td>75</td>
</tr>
<tr>
<td>3.3.2.2</td>
<td>Bottlenecks</td>
<td>81</td>
</tr>
<tr>
<td>3.4</td>
<td>Discussion</td>
<td>83</td>
</tr>
<tr>
<td>4</td>
<td>The Proteomics MIMI</td>
<td>85</td>
</tr>
<tr>
<td>4.1</td>
<td>Introduction</td>
<td>85</td>
</tr>
<tr>
<td>4.2</td>
<td>Development</td>
<td>87</td>
</tr>
<tr>
<td>4.3</td>
<td>Design and implementation</td>
<td>89</td>
</tr>
<tr>
<td>4.3.1</td>
<td>Supporting the experimental workflow</td>
<td>90</td>
</tr>
<tr>
<td>4.3.2</td>
<td>Optimizing scheduling</td>
<td>95</td>
</tr>
<tr>
<td>4.3.3</td>
<td>Tracking research materials</td>
<td>97</td>
</tr>
<tr>
<td>4.3.4</td>
<td>Creating statements and invoices</td>
<td>98</td>
</tr>
<tr>
<td>4.4</td>
<td>Discussion</td>
<td>100</td>
</tr>
<tr>
<td>5</td>
<td>Extreme Programming: A Case Study</td>
<td>101</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>101</td>
</tr>
<tr>
<td>5.1.1</td>
<td>The practices of extreme programming</td>
<td>104</td>
</tr>
<tr>
<td>5.2</td>
<td>The case study</td>
<td>106</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Planning the initial design</td>
<td>106</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Performing the implementation</td>
<td>109</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Using metaphors</td>
<td>112</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Adhering to simplicity</td>
<td>115</td>
</tr>
<tr>
<td>5.2.5</td>
<td>Maintaining a customer relationship</td>
<td>116</td>
</tr>
<tr>
<td>5.3</td>
<td>Discussion</td>
<td>118</td>
</tr>
<tr>
<td>6</td>
<td>Cost-benefit Analysis</td>
<td>120</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>120</td>
</tr>
<tr>
<td>6.2</td>
<td>Cost-benefit Analysis of MIMI</td>
<td>122</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Tangible benefits</td>
<td>124</td>
</tr>
<tr>
<td>6.2.1.1</td>
<td>Administrative workflow tasks</td>
<td>125</td>
</tr>
<tr>
<td>6.2.1.2</td>
<td>Scientific workflow tasks</td>
<td>132</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Intangible benefits</td>
<td>135</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Costs</td>
<td>136</td>
</tr>
<tr>
<td>6.3</td>
<td>Summary of Results</td>
<td>137</td>
</tr>
<tr>
<td>6.4</td>
<td>Discussion</td>
<td>138</td>
</tr>
</tbody>
</table>
7 Conclusions and Future Work

7.1 Conclusions .................................................. 140
7.2 Future work ................................................... 141
  7.2.1 Modifications to improve deployment scope ........ ........ 141
  7.2.2 Linking deployment systems ............................ 144
  7.2.3 caMIMI .................................................. 146
List of Tables

1.1 The estimated number of core facilities in the United States for eight different disciplines ........................................ 13
1.2 A sample of data throughput by equipment at imaging (rows 1-3) and flow cytometry (row 4) core facilities .......................... 16
1.3 The five ideal functional areas of an integrated informatics infrastructure for core facilities ............................................ 24
1.4 Adherence of existing LIMS to the list of ideal tasks ......................... 27

2.1 The number of objects, attributes, and operations that corresponds to using each object model ......................................... 46
2.2 Percentage changes of relevant categories when using the refined object models in comparison to the initial object model, which serves as the baseline .............................................................. 48

3.1 The steps of the data-flow process ........................................... 70
3.2 MIMI’s content object and data statistics .................................. 74
3.3 File numbers which are associated with three imaging systems and used as test cases .......................................................... 75
3.4 Key computers that are part of the CCIR’s network topology, which is illustrated in Figure 3.13 .............................................. 77
3.5 The results of executing the \textit{tracert} command from an imaging system’s workstation PC to measure the number of hops and network latency to the Data Server ...................................................... 82
3.6 The results of executing the \textit{tracert} command from a user’s workstation PC on the CWRU network to measure the number of hops and network latency to the Meta Server .............................................. 82

4.1 The attributes of the study and step objects .................................. 92
4.2 The scheduling algorithm, which is based on dynamic programming ................................................................. 96

5.1 The implementation order for MIMI’s functionalities by object .............. 109
5.2 The metaphorical workflow associations for MIMI’s functionalities by object. Values A, B, and S represent administration, business tasks, and scientific data management, respectively ........................................ 113

6.1 A summary of the cost-benefit analysis of using MIMI at the Case Center for Imaging Research ............................................. 138
7.1  The seventeen cores of the Case Comprehensive Cancer Center . . . . 147
7.2  Examples of objects from caMIMI's data model and the values of their
     attributes, which are expressed as \textit{attribute} : \textit{value} . . . . . . . . . . . . 148
List of Figures

1.1 Total hours spent per week on data handling tasks by sub-discipline and percentage of respondents .................................................... 20

2.1 A biomedical core facility’s key entities ........................................... 39
2.2 Object diagrams for a biomedical core facility’s key common entities . 41
2.3 How OMT defines associations between objects ............................... 42
2.4 The result of applying inheritance to the research material object ....... 45
2.5 The initial object model ................................................................. 47
2.6 The resulting object model after inheritance is applied to the initial object model ................................................................. 48
2.7 The resulting object model after both inheritance and object combination are iteratively applied to the initial object model ................. 49

3.1 The user profile model ................................................................. 58
3.2 UML object diagram of Plone’s objects and their inheritance relationships ................................................................. 59
3.3 Left: Profile request form. Right: Profile management interface, which is used to modify or approve pending profile objects. .................. 60
3.4 The project model ................................................................... 62
3.5 Top: Project request form. Bottom: Project management interface, which is used to modify or approve pending project objects. ............ 63
3.6 The interface for viewing approved project objects and modifying their list of collaborating users ................................................................. 64
3.7 The session model ................................................................... 65
3.8 The scheduling interface .............................................................. 66
3.9 The supplemental billing form, which captures details about a session that has been carried out ......................................................... 67
3.10 The interface for compiling resource usage summary statistics, with example results ................................................................. 68
3.11 The data-flow process, with steps described in Table 3.1 ............... 69
3.12 The interface for downloading scientific data, where a red “X” denotes a file or folder that has been selected for downloading ............. 72
3.13 The CCIR’s network topology. Labels 1-5 are explained in Table 3.4. 76
3.14 The transfer rates for uploading scientific data from imaging systems’ workstation PCs to the Data Server .............................................. 77
### 3.15 The transfer rates for copying scientific data from the Data Server to the Meta Server ........................................ 78

### 3.16 The transfer rates for downloading scientific data from the Meta Server to a user’s local workstation PC on the CCIR network ........... 79

### 3.17 The transfer rates for downloading scientific data from the Meta Server to a user’s workstation PC on the CWRU network .......... 80

### 4.1 The proteomics MIMI’s data model ........................................ 89

### 4.2 The study template model .................................................. 91

### 4.3 The experimental workflow management interface for viewing or modifying existing study templates .......................... 93

### 4.4 A use-case scenario for the study template model .................. 94

### 4.5 The proteomics MIMI’s scheduling interface from a manager’s viewpoint, which is implemented using the Google Web Toolkit ........ 96

### 4.6 A sample invoice for a study in Portable Document Format ........ 99

### 5.1 XP’s high-level relationship between the customer and the developer . 103

### 5.2 Examples of MIMI’s design specifications, which were analogous to “user stories” from XP’s planning game .................... 107

### 6.1 UML diagram of the necessary steps and the corresponding time estimates per working day for using the status quo to subsequently carry out all of the tasks of the administrative workflow ................. 127

### 6.2 UML diagram of the necessary steps and the corresponding time estimates per working day for using MIMI to subsequently carry out all of the tasks of the administrative workflow .................. 129

### 6.3 UML diagram of the necessary steps and the corresponding time estimates per working day for using the status quo to subsequently carry out all of the tasks of the scientific workflow .......................... 133

### 6.4 UML diagram of the necessary steps and the corresponding time estimates per working day for using MIMI to subsequently carry out all of the tasks of the scientific workflow .......................... 134

### 6.5 Net tangible benefits of using MIMI over the first twenty years following the initial launch ............................................ 139

### 7.1 How the centralized communication layer transparently combines multiple deployment MIMI systems .......................... 145

### 7.2 caMIMI’s data model, which follows the specifications of the Unified Modeling Language ............................................. 147
Acknowledgments

This dissertation is a byproduct of the guidance and friendships that I have experienced during the last few years.

I would like to thank my advisor, Dr. Guo-Qiang Zhang for his mentorship and guidance. Throughout the years, he has offered invaluable advice and provided continuous support for my dissertation work.

Thanks also goes out to Dr. Chris Flask, Dr. Michael Branicky, and Dr. Ken Loparo for their time to serve in my dissertation committee and for their valuable, constructive feedback.

A large thank you also goes out to Dr. David Wilson who has provided much support and wisdom and helped facilitate the collaborative work with the Case Center for Imaging Research.

I would also like to thank Adam Troy and the members of the Case Center for Imaging Research for their friendship, support, and collaboration: Dr. Raymond Muzic, Dr. Jeff Duerk, Janet Schiciano, Kelly Covey, and Deborah Sim.

Much appreciation and thanks goes out to Ms. Marisa Tsuji and Ms. Mary Sims for their wonderful friendship and support over the last four years.

Finally, I would like to thank my sister, Dr. Agata Exner, my parents, Dr. Andrzej and Barbara Szymanski, and my brother-in-law, Steve, for their love, support, and guidance – I am lucky to have such a wonderful family.
An Integrated Informatics Infrastructure for Pre-clinical Research-IT Support

Abstract

by

JACEK SZYMANSKI

Modern scientific research is inherently multi-leveled and multidisciplinary. High-throughput scanners and powerful, new generations of equipment operating at core facilities are essential tools for advancing research in disciplines such as bioscience and medicine. They produce large and complex data that must be safely archived and at the same time easily accessible. Together with administrative management tasks, core facilities face a crushing burden in research-IT support.

This dissertation introduces a Multi-modality Multi-resource Information Integration (MIMI) system that combines scientific and administrative support in a seamlessly integrated environment. MIMI captures a complete set of raw information about people, projects, resources, materials, accounting, scheduling, and scientific data. It replaces the existing piece-meal and disintegrated approach to research-IT support by a single comprehensive package. No other system in the market today provides such a complete set of features to meet the informatics infrastructure challenge. MIMI is created from object-oriented data models that represent the flow of information at core facilities and capture a high-level view of their daily operations. MIMI contributes to the design and implementation of interacting workflows that account for the scientific and administrative logics of a core facility. Implemented in the open source content-management system Plone, MIMI has been developed in a style that translates to certain practices of the extreme programming paradigm. MIMI employs the latest techniques in web-based systems such as DHTML, CSS, and AJAX. Additional features include semi-automated dataflow, personalized experimental workflow management, and a dynamic programming approach for optimizing the scheduling of multi-step studies.

MIMI has been deployed at the Case Center for Imaging Research and is being deployed at the Case Center for Proteomics. Cost-benefit analysis shows that substantial financial savings can be accrued long term. Given the magnitude of research performed at core facilities, MIMI not only indirectly impacts research by reducing management overhead, it may also directly impact bioscience and medicine by enabling a more system-oriented approach, now that a greater variety of relevant data and information becomes readily available at the fingertips of researchers.
Chapter 1

Introduction to MIMI

1.1 Core facilities and research-IT

Modern scientific research is inherently multi-leveled and multidisciplinary and occurs across institutes and centers. To facilitate this research, core facilities bring the latest imaging and scanning technologies to the research community. The fundamental enabling technologies for scientific research include specialized equipment such as digital imaging scanners and mass spectrometry machines. With specialized equipment being both prohibitively expensive (scanners easily cost millions) and not portable due to size and weight restrictions, having researchers purchase equipment for individual use is obviously impractical. As a result, core facilities have emerged as hubs for various types of special-purpose equipment, making them accessible to researchers at large. However, core facilities face significant data management challenges unforeseen at their inception. In the midst of increasing pressure from data management tasks, research-IT is recognized as a critical component for advancing
the progress of scientific research [6].

The relationship between core facilities and researchers is classified as “give-and-take.” Core facilities rely on generated revenues to sustain their operation, whereas researchers need core facilities to perform cutting-edge research. The discovery of deoxyribonucleic acid (DNA) by Watson and Crick during the late 1950s had identified the key to life’s inheritance, i.e., “the secret of life [36].” However, the blueprint for the code of life has only been revealed recently through the human genome project, harnessing advances in biological and information technology. The completion of the human genome sequence and the sequencing of additional organisms (e.g., plants, animals, bacteria) open the door for a revolution that will fundamentally change the practice of healthcare using information based medicine [19, 38, 52, 66], stem cell therapy [57, 86], and nanotechnology in medicine [7, 83]. However, all of these promised advances in healthcare cannot be achieved without additional breakthroughs in basic biological sciences, which has led to increasing interest in the subareas of proteomics, flow cytometry, and molecular imaging. What follows is a brief summary of these specialization subareas.

*The proteomics subarea.* The specialization subarea of proteomics studies proteins on cellular and tissue levels to ascertain how their structure, function, and control relates to biological processes [2]. Proteomics researchers often need to analyze large collections of proteins simultaneously following their separation, quantification, and identification. The separation and quantification of proteins are accomplished through two-dimensional gel electrophoresis (applying electric current to a gel matrix
to separate proteins according to isoelectric points and molecular mass) and stain intensities, which subsequently leads to identification using mass spectrometry [73]. The importance of the proteomics subarea is highlighted in the National Institutes of Health (NIH) Roadmap [71], which emphasizes the subarea as a hotbed for biological research that would help: “... to characterize protein function in health and disease and to monitor the markers of a disease in order to deploy early prevention efforts and to identify potential therapeutic targets.” For example, the Fragile X tremor-ataxia syndrome is associated with two proteins: hnRNP A2/B1 and CUGBP1 [87].

The flow cytometry and molecular imaging subareas. A living organism is a complex entity that contains many inter-related components. In order to gain an understanding of a living organism’s functionalities, biological research must be performed on a subcellular level, which brought about the flow cytometry and molecular imaging subareas.

Flow cytometry involves determining physiological measurements on individual cells. It aims for rapid cell analysis and quantification by using equipment that processes microscopic particles which are suspended in fluid. For example, Cappellini et al. [15] used flow cytometry to detect apoptosis-inducing ligand receptors in immature white blood cells of leukemia patients, the results of which were used to evaluate a potential anti-cancer agent.

The molecular imaging subarea refers to investigating the fundamental molecular pathways of a living organism with the use of high-tech equipment across multiple imaging modalities, which include Magnetic Resonance (MR), Position Emission To-
mography (PET), and Ultrasound (US). Molecular imaging is generally performed in a non-invasive manner (it does not rupture an organism’s body cavity) and is used to image drug effects and evaluate disease progression on a molecular level [61].

1.1.1 Types and scope of core facilities

On the highest-level, there are two types of core facilities: pre-clinical and clinical. Core facilities that fall under the clinical designation contain equipment that is used to further research for human studies. These core facilities also must adhere to strict human subject guidelines, which protect the privacy of research patients and sequester their scientific data from unauthorized access. In contrast, pre-clinical core facilities are sites that have equipment which is used to analyze samples such as small animals, liquid solution vials, and tissue cultures. As a result, these core facilities do not have to deal with privacy issues, although adherence to protocol standards such as the approval of samples by the Institutional Animal Care and Use Committee (IACUC) [74] is required. The work of this dissertation focuses on pre-clinical core facilities.

<table>
<thead>
<tr>
<th>Type of core</th>
<th>Estimated number in United States</th>
</tr>
</thead>
<tbody>
<tr>
<td>Medical and Molecular Imaging</td>
<td>150</td>
</tr>
<tr>
<td>Proteomics</td>
<td>162</td>
</tr>
<tr>
<td>Flow Cytometry</td>
<td>155</td>
</tr>
<tr>
<td>Genetics and Gene Expression</td>
<td>53</td>
</tr>
<tr>
<td>Pathology</td>
<td>36</td>
</tr>
<tr>
<td>Microscopy</td>
<td>130</td>
</tr>
<tr>
<td>Tissue Culture</td>
<td>18</td>
</tr>
<tr>
<td>Experimental Data</td>
<td>165</td>
</tr>
</tbody>
</table>

Table 1.1: The estimated number of core facilities in the United States for eight different disciplines
Core facilities span various types of disciplines, resulting in a substantial variety. Table 1.1 shows an estimation of the number of core facilities in the United States for eight different disciplines. The number of Medical and Molecular Imaging core facilities in the United States was estimated in the following manner. Approximately 125 medical schools exist in the United States, almost all of which are associated with an active research component. Moreover, there are many research institutions which do not grant medical degrees but are still involved with imaging research. Examples of such institutions include the Cleveland Clinic Foundation, the main campus of Penn State University, and the University of Illinois at Urbana-Champaign. In addition, pharmaceutical and biotechnology companies as well as government laboratories such as NIH and the Environmental Protection Agency (EPA) also are sites with active imaging core facilities. Assuming that 80% of the 125 medical schools are associated with imaging research and that 50 additional facilities account for the remaining usage, it is estimated that 150 imaging core facilities exist in the United States. Our estimation of the number of core facilities for the other seven disciplines in Table 1.1 used summaries of results from Google search. We utilized queries of the form “X Core Facility” +site:edu, where X is a discipline. For the tissue culture discipline, issuing a query for “Tissue Culture Core Facility” +site:edu returned 29 results which were manually validated, with 18, or 62%, representing unique core facilities. We generalized this percentage to the remaining six disciplines and reduced the number of their Google search results accordingly to compensate for redundant or irrelevant matches. Of course, although Table 1.1 provides reasonable estimates of the number of core facilities in the United States, this number will only increase once future cores
come online.

1.1.2 Core facilities on the CWRU campus

We next describe three types of core facilities on the campus of Case Western Reserve University (CWRU) that are associated with imaging, proteomics, and flow cytometry, respectively. For each type of core facility, we provide a brief overview and then give examples of the corresponding equipment. We also profile the scientific data that are generated to give an idea of the types of scientific data that a core facility handles.

*Imaging core facility.* The Case Center for Imaging Research (CCIR) is a state-of-the-art facility that provides access to the latest in imaging technologies. It spans 6,500 square feet of space and contains shielded equipment rooms and sample preparation areas. The CCIR also employs research assistants, associates, and primary faculty members, who have generated over $20,000,000 of sponsored funds.

The CCIR houses various imaging scanners that allow researchers to obtain scientific results in the form of digital images. These scanners include high-field magnetic resonance (1.5T, 4.0T, 7.0T, and 9.4T), nuclear (MicroPET and X-SPECT), and optical (IVIS 200 and Maestro) imaging systems, which as a whole, span multiple modalities such as MR and PET. What follows are details on three of these imaging scanners, which represent a microcosm of the digital imaging technology that is found at the CCIR: The Siemens Medical Solutions MicroPET, the Bruker Biospec 9.4 Tesla MR, and the Gamma Medica X-SPECT.
The Siemens Medical Solutions MicroPET [49] imaging scanner is based on PET and contains a small horizontal bore, usually eight centimeters in diameter, that holds a small animal or some other sample, which is scanned to generate scientific data. The scientific data that the MicroPET generates consist of three files with different types of extensions [first row of Table 1.2]. An lst file stores the raw scientific data which contain incident information over specific time periods, whereas the hdr and log files are used to interpret these data.

<table>
<thead>
<tr>
<th>system</th>
<th>file format</th>
<th>example size</th>
<th>resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>MicroPET</td>
<td>lst</td>
<td>1.32 GB</td>
<td>2 mm</td>
</tr>
<tr>
<td></td>
<td>hdr</td>
<td>10.07 KB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>log</td>
<td>2.99 KB</td>
<td></td>
</tr>
<tr>
<td>9.4 Tesla MR</td>
<td>fid</td>
<td>23.25 MB</td>
<td>&lt; 1 mm</td>
</tr>
<tr>
<td></td>
<td>2dseq</td>
<td>3.88 MB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>im*</td>
<td>0.13 MB</td>
<td></td>
</tr>
<tr>
<td>X-SPECT</td>
<td>camera</td>
<td>2.00 MB</td>
<td>1/4 mm</td>
</tr>
<tr>
<td></td>
<td>raw</td>
<td>2.00 MB</td>
<td></td>
</tr>
<tr>
<td></td>
<td>slice</td>
<td>0.50 MB</td>
<td></td>
</tr>
<tr>
<td>FACSAria</td>
<td>fcs</td>
<td>40.0 MB</td>
<td>407/488/645 nm</td>
</tr>
</tbody>
</table>

Table 1.2: A sample of data throughput by equipment at imaging (rows 1-3) and flow cytometry (row 4) core facilities

The Bruker Biospec 9.4 Tesla MR imaging scanner generates scientific data that consist of tens to thousands of im files that represent images of a scanned sample [second row of Table 1.2]. The im files are typically generated with respect to total elapsed time or spatial viewpoints. From the second row in Table 1.2, we see that the sizes of im files can be small. However, the corresponding raw data files can be hundreds of megabytes in size and require larger storage space. For example, a researcher who scans multiple rats to obtain scientific data may end-up with > 1 GB
of raw supplemental data to process.

The Gamma Medica X-SPECT imaging scanner utilizes two modalities to generate different types of scientific data. The X-SPECT scanner uses either Single Photon Emission Computed Tomography (SPECT) or Computed Tomography (CT) as a means for acquiring scientific data [82]. Regardless of the acquisition means, the X-SPECT scanner generates scientific data as three types of files that represent visual camera captures [third row of Table 1.2]. These scientific data are in proprietary format and require conversion into a standard visual format by using a vendor-supplied batch processing tool.

Proteomics core facility. The Case Center for Proteomics (CCP) covers an area of 5,000 square feet that is used for laboratory space, sample storage, and equipment housing. The CCP’s mission is: “... to improve human health and the quality of life through the generation, application, and dissemination of new knowledge of biological systems, function, and disease.” The CCP provides resources that help further research in the proteomics specialization area, which is discussed in Section 1.1. It hosts machines including the Finnigan LTQ and DECA-XP Plus, GE 2D-gel DIGE, Applied Biosystems Q-Star, and PerkinElmer proTOF MALDI, with capabilities such as protein separation with two-dimensional gel electrophoresis and protein quantification using mass spectrometry. The scientific data that are generated by proteomics machines are usually < 50 megabytes in size.

Flow cytometry core facility. The CWRU flow cytometry core facility is a part of the Case Comprehensive Cancer Center. It provides researchers with access to
equipment that is used to quantify, sort, and examine microscopic particles which are present in a stream of fluid. The facility is mainly focused on processing particles on a cellular level, and it houses machines including cell sorters (FACSAria and Elite ESP) and analyzers (LSR and Beckman XL) that help accomplish this goal.

The BD Biosciences FACSAria cell-sorting machine contains three lasers that function as high-powered light sources as cells pass through them. These lasers have wavelengths of 407, 488, and 633 nanometers. The FACSAria cell-sorting machine processes approximately 70,000 particle events per second and generates scientific data which contain the results of the analysis in the flow cytometry standard format [fourth row of Table 1.2].

1.1.3 Information and data management challenges

The rapid expansion of research activities at core facilities has brought substantial administrative information and scientific data management challenges. Administratively, a core facility must be able to manage the scheduling of its equipment and to maintain a flexible and effective billing system to track material, resource, and personnel costs and charge for services in order to sustain its operation. It must also have the ability to regularly monitor the usage and performance of its equipment and to gather summary statistics on resources spent on different categories of research for justifying continued funding. In addition, with technology evolving at an incredible pace, the complexity of equipment leads to the creation of a vast profile of scientific data that consist of multiple file formats that occupy varying amounts of storage.
space, as indicated in Section 1.1. As a result, a core facility must not only be able to manage experimental workflow, but also the corresponding set of large and complex scientific data. It must also disseminate experimental data to relevant researchers in a secure and expedient manner that facilitates collaboration and provides support for data interpretation and analysis.

1.1.3.1 The University of Washington study

The informatics challenges at core facilities are carefully analyzed in an extensive survey on research data management, performed by Anderson et al. [6]. This study involved 286 participants from the University of Washington who were recruited via an e-mail list representing a wide range of academic biomedical researchers in the Pacific Northwest. Anderson et al. summarized the data management challenges in the following way: “... investigators are required to study biological problems involving large amounts of diverse data that require special storage and analysis. The use of core facilities has lowered technical barriers and allowed investigators to generate and collect data outside their own discipline. Yet despite these advances, widespread access to highly detailed, heterogeneous research data has dramatically increased the difficulties faced by investigators ...” The survey results reported that 84% of the respondents experienced issues with data management, with many spending > 10 hours per week on data handling tasks [Figure 1.1], which caused a backlog in laboratory productivity. Software solutions to address these issues were either too expensive and complicated or not available at all, with only 14% of respondents using some type of data management tool. In addition, although general-purpose applications (e.g.,
spreadsheets or text files) were often utilized, 60% of respondents noted that these applications were limited by size constraints and lack of processing power.

![Figure 1.1: Total hours spent per week on data handling tasks by sub-discipline and percentage of respondents](image)

The results of the survey also confirmed that the difficulties in managing scientific data involve data storage, organization, and retrieval, as well as the need to improve the existing methods for handling large datasets, annotating data, and sharing results. Long term archiving of scientific data was a major concern with 28% of the respondents. Issues were noted with increases in the number of files and their corresponding sizes in addition to the proliferation of complex and varied file formats. Also, many respondents utilized their own methods for organizing data and faced problems including truncated filenames and the absence of nomenclature for data sharing. Some respondents even followed no specific method for organizing their data: “Data are not organized in any way – they are just thrown into files under different projects.”
The University of Washington study concluded that the current approaches for managing scientific data prior to analyzing and processing need to be revamped. The ad hoc approaches to data organization that are dealt with by individuals should be replaced with a standardized organization method. With financial and time constraints burdening the individual from acquiring expertise and tools, the study pointed to informatics support on a core facility level. Anderson et al. also concluded that the current state of institutional data management and analysis support is inadequate and recommended developing information service cores that can address the specific needs of researchers. Such service cores would use a common information support infrastructure across domains and feature modern data exchange models and formats.

1.1.3.2 Deficiencies in existing infrastructures

Although core facilities address some of their informatics challenges using existing infrastructures, our own analysis of informatics at the existing CWRU core facilities concludes that deficiencies are manifested in multiple areas.

Substantial administrative and personnel overhead. This type of overhead exists in pen-and-paper based record keeping aided by disconnected spreadsheet programs, manual management of scheduling on a common off-the-shelf calendar system that operates in isolation, using portable media for data transport, and relying on email communication to gather a variety of project related information. Some core facilities operate under an IT (Information Technology) infrastructure resulting from
adopter/adapting existing open-source/in-house/commercial software for managing a variety of data, though this only reduces the problem to the equally, if not more, challenging issues of information integration, interoperability, and resource usage for IT personnel support.

Lack of support for collaboration among researchers. The disintegration of administrative and scientific data makes it difficult to access data and find information about related prior studies, which is confirmed by Anderson et al. [6], who noted that limited search and retrieval capabilities were common with disorganized data. Collaborating researchers must then rely on ad hoc mechanisms such as email communication to share data and results. This not only makes the book-keeping of data a chore, but also lacks a uniformly enforceable standard for the safety of valuable data and results from analyses.

Significant amount of redundant, disintegrated, and inconsistent data. When data are kept in disconnected systems, information such as a principal investigator’s contact profile and projects may have to be re-entered multiple times to multiple systems, making it difficult to maintain and update. Repetition in data entry not only requires additional effort, but also opens more room for errors and inconsistencies. The same entities may have been entered using different names in different systems, and changes made in one system may not automatically propagate to other systems. This is also highlighted by Anderson et al. [6].

Lack of support for the integration of information from disparate resources. Access to data and knowledge is often labor-intensive, repetitive, disorganized, and
burdensome; project management and data analyses are tasks relegated to individual investigators without a common framework or standard for record-keeping or for sharing and collaboration using intermediate results.

The root cause for these deficiencies can be summarized as a lack of a holistic approach to infrastructure support, which further stresses the need for a software solution that can address the corresponding administrative information and scientific data management challenges of core facilities.

1.1.3.3 An ideal informatics infrastructure for research-IT

An integrated informatics infrastructure for core facilities would consist of various specialized software and hardware, which would work in-tandem to effectively support aspects of administrative information and data management. Such an infrastructure would acquire details about a core facility’s workflow, samples, or scientific data and present these specific details to both researchers and core facility managers. It would also be comprised of a database back-end (e.g., MySQL) and an interactive front-end for the user.

To address the existing informatics challenges (for example, see Section 1.1.3.1), an integrated informatics infrastructure for core facilities would ideally cover five functional areas: managing administrative information and scientific data, analyzing scientific data, supporting laboratory workflows, supporting core-specific tasks, and managing the hardware and software components [Table 1.3]. The scope of these areas can be different across all types of core facilities. Some core facilities place more
emphasis on gathering a plethora of information and scientific data, while others may
focus more on analysis since they are not inundated with a significant amount of
information and scientific data. For example, an active flow cytometry core facility
(see Section 1.1.1) that is responsible for processing and tracking 1,000 small tissue
samples on a daily basis will spend more time in the data capture phase than a low-
volume core facility that deals with only 1,000 total samples annually. Of course,
differences between functional areas not only abound across the same types of core
facilities, but also appear across core facilities of various disciplines such as imaging
and proteomics.

<table>
<thead>
<tr>
<th>functional area</th>
<th>tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managing administrative information and scientific data</td>
<td>- manual information entry</td>
</tr>
<tr>
<td></td>
<td>- manual data transfers</td>
</tr>
<tr>
<td></td>
<td>- two-way communication for automated data transfers</td>
</tr>
<tr>
<td>Analyzing scientific data</td>
<td>- associations between multiple scientific data</td>
</tr>
<tr>
<td></td>
<td>- automated creation of analysis reports</td>
</tr>
<tr>
<td>Supporting laboratory workflows</td>
<td>- experimental workflow management</td>
</tr>
<tr>
<td></td>
<td>- semi-automated scheduling of studies</td>
</tr>
<tr>
<td>Supporting core-specific tasks</td>
<td>- tracking samples for inventory assessment</td>
</tr>
<tr>
<td></td>
<td>- tracking costs and revenues</td>
</tr>
<tr>
<td></td>
<td>- billing and accounting</td>
</tr>
<tr>
<td></td>
<td>- automated creation of assessment reports</td>
</tr>
<tr>
<td>Managing hardware and software components</td>
<td>- archiving scientific data</td>
</tr>
<tr>
<td></td>
<td>- using redundant storage</td>
</tr>
<tr>
<td></td>
<td>- optimizing packet size for data exchange</td>
</tr>
</tbody>
</table>

Table 1.3: The five ideal functional areas of an integrated informatics infrastructure for core facilities

1.1.3.4 Related work

Examples of Laboratory Information Management Systems (LIMS) that focus pri-
marily on the functional area of managing scientific data have been the subject of
investigation since the last decade of the twentieth century. An important trend that
attempts to address the challenge of scientific data management is a shift towards web-based LIMS that are conducive to location-independent access.

Dai et al. [21] and Jakobovits et al. [39] described web-based repository management systems, while Cogno et al. [18] and Tews et al. [94] elaborated on web-based systems for distributing biomedical information. Others have introduced the Web Interface Repository Manager (WIRM) that behaves as an experiment management system, facilitates distribution of multimedia biomedical data over the web, lowers the overall cost of knowledge sharing, features a layered database approach, and consists of a variety of free software [40, 41, 42, 43].

For clinical settings, Deo et al. [22] created a web-based system for managing information about diabetes patients that uses various components such as an Oracle repository. Nguyen et al. [70] utilized an open-source, web-based application called DADOS-Prospective to disseminate clinical data in spreadsheet format via the internet. Others have introduced web-based clinical repositories for accessing scientific data such as the results of radiation therapy [23, 60, 84, 103].

For the general small-scale biomedical research laboratory, Fong et al. [30] introduced a customizable electronic laboratory online (CELO) that allows biomedical research laboratories to securely search and share scientific data over the internet. Furthermore, Keller et al. [48] and Li et al. [58] have elaborated on web-based laboratory information systems that feature the required breadth of functionalities to address the scientific data management needs of various laboratories.

Although there have been many examples of LIMS that emphasize scientific data management, examples of LIMS that are optimized to support the workflows of lab-
oratories also exist. Assimakopoulos et al. [8] described the Workflow Management and Systems Approach (WOMASA), which promotes general workflow management for scientific laboratories by introducing a platform for both workflow specification and execution. Goodman et al. [33] elaborated about the LabBase system that provides a well-defined workflow model for laboratory settings with the use of concepts such as Materials, Steps, and States. Wade et al. [98] even described combining workflow management systems and distributed systems to develop a framework for open-laboratory computer services.

1.2 Problem statement

Previous work on laboratory information management systems has been mainly devoted to web-based scientific data management and workflow management for scientific and clinical laboratories, as described in Section 1.1.3.4. However, the marketplace currently lacks LIMS that meet the informatics requirements of the general core facility. There are no LIMS for core facilities that comprehensively meet the functional areas which appear in Table 1.3 and represent an ideal platform for research-IT support.

The weaknesses/shortcomings of existing LIMS can be seen in Table 1.4. The existing LIMS fall short in supporting two key functional areas: laboratory workflows and core-specific tasks. These LIMS are primarily concerned with facilitating the scientific data acquisition process for the researcher and do not provide core facility features such as scheduling studies and tracking costs and revenues. In light of the
Table 1.4: Adherence of existing LIMS to the list of ideal tasks

drawbacks of existing LIMS, this thesis describes our work over the past several years on developing a common, comprehensive informatics infrastructure that is applicable across a wide-range of core facility frameworks and facilitates research through a high level of data integration.

An informatics infrastructure for core facilities is a challenging system to develop, but developing such a system may introduce methods for furthering research within a large-scale, integrated environment. Such a system would also increase collaboration among researchers [6]. In addition, given the challenges encountered by imaging and other kinds of core facilities, an approach that captures a vision for a long-term solution and addresses some of the immediate needs is desirable.
1.3 Contributions of this thesis

To facilitate information and scientific data management across multiple core facilities, this thesis introduces an integrated informatics infrastructure known as the Multi-modality Multi-resource Information Integration (MIMI) [89, 90, 91, 106] environment for such facilities. Contributions of the thesis include a common integrated informatics infrastructure for cores, a set of comprehensive features in a single system, and details on data modeling, system development, and deployment.

1.3.1 A common informatics infrastructure for cores

MIMI combines the management of a core facility’s scientific and administrative workflows in a package that has a tangible impact on daily operations.

The scientific workflow involves handling scientific data and disseminating them to researchers. The administrative workflow deals with scheduling equipment for use, billing, accounting, and assessments of equipment.

MIMI can easily adapt across multiple core facilities and other similar information centers and can apply with little modification to the scientific and administrative functionalities that are common across environments of different breadths and sizes. It can also alleviate the data fragmentation and support headaches that appear when dealing with software solutions from multiple vendors.
1.3.2  A set of comprehensive features in a single system

MIMI integrates administrative and scientific functions of a core facility and captures a complete, essential set of raw information about people (researchers and staff members), projects, resources, samples, accounting information, billing details, scientific data, and resource scheduling. It replaces existing piece-meal research-IT support in one comprehensive and integrated package – no other software in the market today provides such a complete set of features.

MIMI provides a semi-automated data workflow for scientific data. It links scientific data to resource scheduling through the use of a platform-independent software application that transfers these data from a resource’s workstation computer to a unique storage location. The storage location is represented by a unique path string that is created during the process of resource scheduling.

MIMI also features minimal overhead for data entry and administrative support. This is achieved by decentralized content management that eliminates redundancy through the use of extensive content object relationships. MIMI manages the administrative workflow by tracking aspects such as cost and staff times spent on preparation and data acquisition, and it helps automate the billing process through automated cost calculation and account linkage. It also provides online monitoring of the overall resource usage of a core facility by compiling on-the-fly summaries such as a profile of usage statistics of equipment and types of involved projects.

MIMI also supports a dynamic programming approach for scheduling multi-step studies with optimized resource usage.
1.3.3 Data modeling, system development, and deployment

MIMI is created from object-oriented data models that represent the flow of information at core facilities. These data models capture a high-level view of the daily operations of core facilities into a conceptual design.

MIMI utilizes cutting-edge technology in software engineering such as AJAX, Javascript, DHTML, and CSS to lower maintenance requirements and provide a web-based portal interface that promotes wide-scale accessibility that is unconstrained by space, time, or type of computer system. MIMI is implemented completely using open-source software, which provides a low developmental cost and promotes system modifiability. It is also a byproduct of a developmental methodology which is a natural fit to certain practices of extreme programming.

MIMI is successfully deployed at two core facilities: The Case Centers for Imaging Research and Proteomics. MIMI allows managers to approve users and projects, schedule resources, manage invoices, perform site-wide search, and archive/backup scientific data. MIMI also permits researchers to create research projects, schedule resources, download and view scientific data, monitor charges incurred, track research samples, and manage the sharing of scientific data, all through a single web-based entry point.

1.4 Thesis overview

We will now provide an overview of the remaining chapters of the thesis. Chapter 2 introduces three object models that represent the relationships between the key
common entities of core facilities. We compare two of the object models to an initial object model to assess their effectiveness when translated into a functional implementation. This chapter is the “building block” for the blueprints to creating MIMI. Chapter 3 summarizes our developmental and deployment experiences with MIMI for the Case Center for Imaging Research, which is an imaging core facility. It describes the guiding principles used during development in addition to the system design, which elaborates on most of the features described in Section 1.3.2. This chapter also profiles the usage of the system and assesses its performance with details on transfer rates and bottlenecks. Chapter 4 elaborates on a deployment MIMI system for the Case Center for Proteomics, a proteomics core facility. It provides details about development and contrasts the system with the Imaging MIMI described in Chapter 3 to highlight key differences. For example, the Proteomics MIMI expands upon the Imaging MIMI by using a dynamic programming approach for scheduling multi-step studies. Chapter 5 details how the developmental methodology for the Imaging MIMI is related to using specific practices of extreme programming. This chapter also serves as a guiding tool for what to expect during development when choosing to create an integrated informatics infrastructure that has parallels with MIMI. Chapter 6 outlines the benefits and costs of using the Imaging MIMI at the Case Center for Imaging Research (CCIR). To estimate the benefits, this chapter uses the the status quo (i.e., prior practice) at the CCIR as a base-line. Performing a comprehensive cost-benefit analysis of using MIMI at the CCIR gives core facilities a substantiation of the financial side, which can translate to their corresponding deployment environments. Core facilities can receive sufficient evidence of tangible benefits such as knowing whether
development and implementation costs are going to be offset in the future. Chapter 7 provides conclusions and suggests future work that relates to MIMI.
Chapter 2

Data Modeling of a Core Facility’s Information Flow

2.1 Introduction

As the complexity of software increases, it is imperative to begin the developmental process by defining a correct set of data models, since design errors which occur during the modeling stage are ten to one-hundred times more expensive to correct at later stages of development [59]. In order to facilitate not only defining data models, but also carrying out the remaining stages of development, many software developmental methodologies have appeared in the last few decades. These software developmental methodologies tried to bring about innovations in the software developmental process to address the inherent difficulties of software development. Throughout the early years, object-oriented software development has evolved alongside object-oriented programming languages such as Python. It has separated itself
from the pool of software developmental methodologies to become a highly prevalent approach to software development.

Object-oriented modeling is an essential component to all object-oriented software developmental methodologies. The purpose of object-oriented modeling includes mediating the process of formulating a correct set of data models for a specific software system through the use of abstract objects. Object-oriented modeling is attractive to the developer for two reasons. Working with a collection of objects is an intuitive experience for the developer because objects are representations of entities such as people. Objects also remain static throughout the development process and are thus easier to manage during a complex development cycle. There exist many examples of object-oriented modeling approaches. Coad and Yourdon [16] described a modeling approach that uses templates and components, while Wirfs-Brock et al. [101] introduced collaborations and hierarchies into the object-oriented modeling domain. In addition, Graham [35] and Swatman [88] elaborated on layer and structural models, respectively, while Wieringa [100] surveyed object-oriented specification methods. Examples of applying object-oriented modeling in different disciplines are found in the literature and corroborate the overall attractiveness of the approach. Booth [12] used an object-oriented approach to model a factory environment within a flexible manufacturing setting, while Campbell and Cheng [14] worked with an object-oriented model for a telemedicine application.

An object-oriented approach of particular interest is the object modeling technique, or OMT [77]. OMT is a method for defining both the static and dynamic entities of a system through the use of objects. OMT specifies three models: The
object model, the functional model, and the dynamic model. The object model deals with the static entities of a system and is the most critical since it acts as OMT’s cornerstone. The functional and dynamic models are more concerned with a system’s behavioral aspects. Results using OMT include various undertakings. Zhang et al. [104] modeled histories of information retrieval queries, while Umeda [96] modeled a manufacturing enterprise system. Sahraoui and Blum [79] even superimposed OMT onto a software life-cycle process model.

This chapter, which describes the first step in the software developmental process, elaborates on using OMT to model a core facility’s information flow with three different object models (behavioral aspects are modeled in Chapter 3) that represent the key common entities of core facilities: an initial model and two other models that are iterative refinements of the initial model. In Section 2.2, we outline how preliminary information was gathered before building the object models. We also describe how the object models were built and explain how the effectiveness of the other two object models was assessed in comparison to the initial object model, the latter being important for code refactoring and size. In Section 2.3, we introduce the three object models and elaborate on the results of the effectiveness assessment. In Section 2.4, we offer concluding remarks.
2.2 Methods

2.2.1 Gathering preliminary information

We used a combination of meetings and shadowing experiences to obtain information that served as a comprehensive basis for building the object models. We held regular meetings with personnel to provide insight into the daily operations of biomedical core facilities. These meetings frequently included staff members who provided an overview of specific core facility procedures. For example, Dr. Raymond Muzic, a modality expert, elaborated on generating scientific data, and Dr. Chris Flask, a facility manager, outlined accounting and billing procedures. We found that the daily operational routine of a typical biomedical core facility features two stages.

Scheduling of resources. The first stage of the daily operational routine begins when a researcher contacts the staff members at a core facility and requests to schedule a resource. Independent of the type of request, the staff members proceed to capture the researcher’s contact information (e.g., phone number and email address) and project details (e.g., what kind of research being performed and under which grant, if any) using either paper records or spreadsheet applications such as Microsoft Excel. These details are then used to schedule a specific resource, which is typically done with an online calendaring system.

Generating scientific data and billing statements. A scheduled resource is typically used to generate scientific data, which as a byproduct leads to the creation of a billing statement by a staff member. Using a scheduled resource is tantamount to
incorporating research materials as an input source and processing them to generate scientific data that are important to the researcher. After a scheduled resource has been utilized, the staff members record relevant information such as the exact duration of use and the types of applicable research materials that were processed. They then proceed to calculate the total cost of use, which is incorporated into a billing statement that can be forwarded to the researcher.

Once we achieved a general understanding of the daily operations of a biomedical core facility, the next step involved gaining “hands-on” experience by shadowing various personnel at a biomedical core facility.

We studied the routines of biomedical core facility personnel on a weekly basis for about two months. The goal was to obtain first hand experience about some core facility aspects that were not outlined during the meeting phase and the corresponding daily operational routine. For example, specific paper records were retrieved and analyzed.

2.2.2 Building the initial object model

In forming the initial object model that covers the key entities of biomedical core facilities, we followed two specific techniques.

Object modeling technique. The object modeling technique (OMT) stands as a time-tested approach for effective system modeling. OMT defines the object model, which encapsulates the static aspects of a system by using a high-level, abstract viewpoint that is easily understood. It contains objects and describes their names,
attributes, operations, and relationships to other objects.

Abstraction. When developing an object model, it is necessary to have an understanding of the term abstraction to see the rationale behind choosing specific objects. The notion of abstraction involves recognizing the relevant entities about something and ignoring the unimportant ones. Following the abstraction process’s selective mind frame allows a developer to build a model that is concise and to-the-point and is applicable to a large pool of systems or places.

2.2.2.1 Object identification

After receiving a comprehensive overview of the daily operations of biomedical core facilities through meetings and shadowing experiences, we applied the abstraction principle and identified the following key common entities that will be used as objects for the initial object model [Figure 2.1].

People. Biomedical core facilities are hubs for people. These people generally divide into four groups: managers, equipment operators, principal investigators, and collaborators. Managers are responsible for efficiently running core facilities. Their role in the daily operational routine involves acting as a core facility’s “staff.” Equipment operators troubleshoot and operate equipment such as medical imaging devices. Their use is evident primarily in the second stage of the daily operational routine, where they take the place of someone who does not have operating experience with a specific type of equipment. Principal investigators perform research and are a core facility’s primary “customer,” while collaborators work with principal investigators
Figure 2.1: A biomedical core facility’s key entities and are affiliated with research projects.

*Scientific data.* Examples of scientific data include medical images, micro-array data, 2D-gel mappings, and raw data. Scientific data at inception are typically a binary data stream that travels from a piece of equipment to a workstation computer via a dedicated connection. These data transform into a human-friendly format on the workstation computer: a collection of binary files inside a folder hierarchy. Scientific data can change storage formats when transitioning into the hands of the appropriate personnel. Two popular approaches for this task are copying data onto portable media such as Digital Versatile Discs or enabling file-sharing on a local area network. The file-sharing approach is similar to the portable media approach, but it replaces the destination media with a computer accessible via a local area network: scientific
data reside inside a shared folder on a workstation computer and are downloadable
from multiple personal computers. *We treat scientific data as a consistent entity
when transitioning from a workstation computer to portable media or file-sharing, so
scientific data are not duplicated.*

**Research materials.** The input source for scientific data generation is research ma-
terials, which include small animals such as rats or woodchucks. Cell plates or liquid
solution vials are additional examples. Some research materials have common aspects
that can categorize them under one category. For example, a principal investigator
may own woodchucks from the same vendor with identical strains.

**Equipment.** Equipment generates scientific data and is a biomedical core facility’s
staple of attraction. One piece of equipment such as the Bruker Biospec 9.4 Tesla MR
system can cost in excess of $1,000,000. Equipment incurs substantial maintenance
fees and hence does not have a fixed cost. Each piece of equipment is generally
connected to a workstation computer that functions as the temporary storage medium
for scientific data. Some equipment processes research materials in parallel to reduce
operational time. For example, the Xenogen IVIS 200 imaging system features a
housing unit that accepts up to five small animals.

**Administrative support.** Biomedical core facilities manage administrative aspects
that include research projects, sessions, and billing statements. Research projects
typically serve as validations of successful grant applications. They also generally
specify a principal investigator as their owner in addition to listing collaborators who
can provide useful expertise. Sessions represent scheduled resources (i.e., equipment)
during a set date and time. Billing statements are session cost summaries and include
the total session cost.

Figure 2.2: Object diagrams for a biomedical core facility’s key common entities

2.2.2.2 Object names, attributes, and operations

OMT requires object specification with an object diagram. Each object diagram
represents a distinct object type. The object’s name appears in the diagram’s top
section, and its attributes appear in the middle section. The bottom section holds the
operations that an object can perform such as setting the values of attributes with the \textit{set} method. Figure 2.2 shows the object diagrams for the objects that represent key common entities of biomedical core facilities. The level of abstraction in this instance introduces only the most pertinent attributes and operations. For example, we see that the research material object contains an “identification tag” attribute. This attribute serves as a unique identifier for a research material to facilitate future referencing tasks.

\subsection*{2.2.2.3 Object associations}

OMT defines associations [Figure 2.3] between objects as follows: A solid line represents a “one-to-one” association, while a solid line that ends with a solid black dot specifies a “one-to-many” association, with the black dot denoting “many.”

![Figure 2.3: How OMT defines associations between objects](image)

We will now describe the associations between the identified objects for key common entities of biomedical core facilities to completely define the initial object model. Principal investigator objects have multiple associations with research project objects because principal investigators can work on many research projects during their research careers. Research projects also require reserving equipment time to generate scientific data. This leads to multiple associations from a research project object to session objects. Associating a research project object with multiple collaborator ob-
jects accounts for a research project having joint collaborators. In addition, placing multiple associations between a principal investigator object and research material objects shows that a principal investigator can own different research materials. A session object has the most intricate associations. It associates to multiple equipment operator objects since at least one person must operate a piece of equipment. A session object also associates with multiple research material objects to highlight that research materials are typically a part of using equipment time. A session content object also contains single associations to a billing statement object and a scientific data object. The former association accounts for the cost of using equipment time. The latter one shows that using equipment time generates scientific data. The scientific data object associates with an equipment object to represent that scientific data originate from equipment.

2.2.3 Refining the initial object model

We refined the initial object model with the intent of improving its effectiveness by forming two additional object models that cover the key entities of biomedical core facilities. We applied the following techniques.

*Inheritance.* The term inheritance is applicable to the objects in a model. It involves taking common attributes and operations among objects and placing them into a super-object. The super-object acts as the parent and “hands-down” any common attributes and operations to its sub-objects. The main reasons for using inheritance are to (1) reduce coding redundancy and (2) maintain coding integrity.
Using inheritance eliminates attribute and operation replication over multiple objects. It also facilitates coding changes since updating a super-object is tantamount to updating multiple objects without the use of inheritance.

Object combination. The term object combination is the process of combining multiple objects into a single object. This process applies to situations where the sub-objects of a super-object parent share identical attributes but have different values for these attributes. It involves transferring these attribute values into a list that becomes a single attribute of the super-object parent. Object combination also applies when it is logical to combine multiple objects with different attributes into a single object. The resulting object then harvests all of the attributes from multiple objects. The motivation for using object combination involves (1) reducing system overhead and (2) maximizing storage space.

2.2.3.1 Applying inheritance

We will apply inheritance to the initial object model. The research material object is a candidate for inheritance. It contains the attributes type, vendor, housing location, and description that can be a part of a super-object. The super-object then becomes the parent of multiple sub-objects that form a common collection of research materials. All research material sub-objects do not share the same parent super-object since research materials have different types. For example, a collection of woodchucks does not have the same parent super-object as a collection of cell plates.

OMT denotes inheritance by placing a triangle above associations between a super-
Figure 2.4: The result of applying inheritance to the research material object and its sub-objects. Figure 2.4 displays the result of applying inheritance to the research material object.

2.2.3.2 Applying object combination

We will now continue iteratively refining the initial object model by applying object combination. We note that the research material super-objects and their sub-objects are candidates for object combination. The value of each sub-object’s identification tag attribute transfers to a list that becomes an attribute of the super-object. Sub-objects with identification tags material 1, material 2, and material 3 combine into a super-object with an attribute value of [ material 1, material 2, material 3 ].

We find that the session, billing statement, scientific data, and equipment objects are also candidates for object combination. We will combine these three objects into one object by transferring their attributes and operations to the session object.
2.2.4 Assessing effectiveness

To ascertain the effectiveness of the other two object models in comparison to the initial object model, we assumed that a core facility will be associated with approximately 2,500 distinct research materials (spanning 500 collections with 5 research materials per collection) and 10,000 sessions during its lifetime and then measured the number of objects/attributes/operations that correspond to these estimates when using each of the three object models [Table 2.1].

<table>
<thead>
<tr>
<th></th>
<th>initial model</th>
<th>inh model</th>
<th>inh-oc model</th>
</tr>
</thead>
<tbody>
<tr>
<td>research material objects</td>
<td>2,500</td>
<td>3,000</td>
<td>500</td>
</tr>
<tr>
<td>research material attributes</td>
<td>25,000</td>
<td>9,000</td>
<td>5,000</td>
</tr>
<tr>
<td>and operations</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>other relevant objects</td>
<td>40,000</td>
<td>40,000</td>
<td>10,000</td>
</tr>
<tr>
<td>other relevant attributes</td>
<td>200,000</td>
<td>200,000</td>
<td>180,000</td>
</tr>
<tr>
<td>and operations</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2.1: The number of objects, attributes, and operations that corresponds to using each object model.

We found that using the initial object model results in 2,500 research material objects in addition to 10,000 session, billing statement, scientific data, and equipment objects (40,000 total). Since the research material object contains 10 attributes and operations, and the session, billing statement, scientific data, and equipment objects correspond to 20 total attributes and operations, we calculated that the number of attributes and operations equals 25,000 and 200,000, respectively.

In addition, using the resulting object model (a.k.a., the inh model) after inheritance is applied to the initial object model gives the same number of session, billing statement, scientific data, and equipment objects/attributes/operations as utilizing
the initial object model. However, it also results in 3,000 research material objects (500 super-objects and 2,500 sub-objects), which translates to 9,000 total attributes and operations (500 x 8 + 2500 x 2).

Also, we found that using the resulting object model (a.k.a., the inh-oc model) after both inheritance and object combination are applied to the initial object model results in 500 research material objects and 10,000 total session objects, which translated into 5,000 and 180,000 total attributes/operations.

### 2.3 Results

![Diagram](image)

Figure 2.5: The initial object model

Figures 2.5-2.7 display the object models for the key common entities of core facilities. Figure 2.5 is the initial object model. Figure 2.6 shows the resulting object model after inheritance is applied to the the initial object model. Figure 2.7 displays the resulting object model after both inheritance and object combination are
iteratively applied to the initial object model.

![Object Model Diagram](image)

**Figure 2.6:** The resulting object model after inheritance is applied to the initial object model.

<table>
<thead>
<tr>
<th></th>
<th><strong>inh model</strong></th>
<th><strong>inh-oc model</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>research material objects</td>
<td>+20%</td>
<td>-80%</td>
</tr>
<tr>
<td>research material attributes and operations</td>
<td>-64%</td>
<td>-80%</td>
</tr>
<tr>
<td>other relevant objects</td>
<td>0%</td>
<td>-75%</td>
</tr>
<tr>
<td>other relevant attributes and operations</td>
<td>0%</td>
<td>-10%</td>
</tr>
</tbody>
</table>

**Table 2.2:** Percentage changes of relevant categories when using the refined object models in comparison to the initial object model, which serves as the baseline.

Table 2.2 summarizes the effectiveness assessment of the latter two object models in comparison to the initial object model. We find that using the resulting object model after inheritance is applied to the initial object model leads to a 20% increase in the number of research material objects but a 64% decrease in the corresponding......
number of total attributes and operations. Furthermore, we see that using the resulting object model after both inheritance and object combination are applied to the initial object model provides even greater overall percentage reductions.

![Figure 2.7: The resulting object model after both inheritance and object combination are iteratively applied to the initial object model](image)

**2.4 Discussion**

This chapter introduced three object models for the key common entities of a biomedical core facility. These object models were the result of partially applying the object modeling technique (OMT) along with the principles of inheritance and object combi-
bination. Based on the results of an effectiveness assessment, the object model that results from applying inheritance and object combination to the initial object model serves as the most representative blueprint for step one of the object-oriented software development of an infrastructure that addresses the needs of a biomedical core facility. It also acts as a reference that is well-defined and can be used universally to understand the aspects that a core facility works with on a daily basis.
Chapter 3

The Imaging MIMI

3.1 Introduction

Many procedures in different types of environments are repeated multiple times on a daily basis. For example, workers on a car assembly-line integrate parts such as wheels and tires in a specified order, and applying for a loan in person involves going to a bank and retrieving, completing, and submitting the relevant paperwork. Independent of the type of procedure, the common thread for all procedures is that they can be partitioned into a collection of well-defined, concise steps. These steps have interdependencies on other steps, where certain steps cannot be carried out before others. For example, a scientist at an imaging core facility cannot retrieve scientific data before they are generated by a scanner. A procedure’s collection of well-defined steps in-tandem with the tangible interdependencies is known as a workflow [32].

With the existence of multiple workflows of varying complexity across many disciplines, Workflow Management Systems have appeared as mediators for both specify-
ing and carrying out workflows, which is evidenced by examples from the literature. For example, Cai et al. [13] described a workflow management system known as Dartflow, which uses web-based technology and transportable agents to improve workflow integration for the business environment. In addition, Miller et al. [67] elaborated on the Webwork component of METEOR-2, which acts as a web-based workflow management system that facilitates commands, communication, and controls for workflow steps. Also, Kang et al. [44] illustrated a multilevel secure workflow management system which is applicable to high-security environments such as the Department of Defense, while Agostini and de Michelis [3] introduced a workflow management system, which emphasized the use of simple process models.

While existing Workflow Management Systems are used to specify and carry out a variety of workflows, they lack the ability to handle different core facility workflows with cross-dependencies that depend on the completion of steps that are a part of other workflows. This chapter describes a system called MIMI (Multi-modality Multi-resource Information Integration environment) that supports both the administrative and scientific workflows of a core facility, which are interacting workflows with cross-dependencies. MIMI not only provides a flexible and expandable solution to the workflow management needs mentioned above, but also provides a foundation for a more advanced system which substantially integrates existing knowledge with analyses and curation of experimental data. We report the design, development and deployment experiences of a baseline Imaging MIMI system at the Case Center for Imaging Research (CCIR), a typical large-scale imaging core facility.

The Imaging MIMI system comes with a web-based interface to support core
membership and project information management. It features an expandable and
modifiable framework that can adapt to the needs of imaging and other kinds of core
facilities. In Section 3.2, we describe our developmental guidelines and MIMI’s system
design and implementation. In Section 3.3, we present usage statistics automatically
gathered by MIMI’s built-in tracking tool. We also provide a performance analysis in
order to demonstrate transfer rates and bottlenecks. In Section 3.4, we give concluding
remarks.

3.2 Methods

3.2.1 Guiding principles and approach

In developing MIMI and its particular realization for imaging research, we have ad-
hered to the following set of guiding principles:

(1) Use an open-source environment for development

(2) Fully integrate the biomedical enduser into the developmental team

(3) Maintain uniformly web-based, menu-driven, friendly user-interface

(4) Decentralize data and information management tasks with role-based access
    control, semi-automated data-flow and resource scheduling to minimize over-
    head after deployment

(5) Employ the latest methodologies and tools in information technology and soft-
    ware engineering in software development
The choice of an appropriate open-source developmental environment [63] not only saves developmental cost, but also ensures that the system is modifiable and expandable without proprietary restrictions. The potential downside of a steeper learning curve and the stability of the supporting community can be overcome by a careful scrutiny of the available open-source packages and suitable training of the programmers. MIMI uses Plone [95], which is an open-source content management system, as its main developmental environment. We chose Plone for its web-based interface for development and its built-in web-server incorporating the latest techniques for content-management, such as version control and cascading style sheets (CSS [65]). Plone’s object-oriented framework allows rapid development through code reuse and extension of proven functional modules. The object-oriented paradigm allows objects placed inside other objects (such as folders) to inherit and reuse their parents’ attributes, contents, and functions. Plone’s object-oriented framework extends to the storage level, allowing developers to conceptually organize information in a logical manner which in turn speeds-up development. The Plone distribution is available for major operating systems such as Mac OS, Windows, and Linux, so a developer can select a preferred environment for development. Our earlier successful experience with Plone [105] also made it an easy choice.

Fully integrating the enduser into the developmental team ensures usability, relevance and impact to the targeted application domain. Although neither consciously nor strictly following the Extreme Programming [17] practice, we find it extremely important to engage the enduser into all steps in the software development process. The engagement of the enduser helps realizes two of the core values of Extreme Pro-
gramming immediately: communication and feedback. Through regular meetings, ongoing changes to loosely specified requirements occur as a natural process. The adaptability to changing requirements is a more realistic and better approach than attempting to define all requirements at the beginning of a project, because the developer and the enduser rarely have complete foresight of the desired end product at its inception. Rather, the ongoing discussions become a cooperative activity that helps define, refine and deepen the understanding of what is desired. However, discussions alone without a concrete system would not be effective. This brings us to the second aspect related to Extreme Programming: test-driven development. Although the goal of test-driven development is to make sure that current code meets requirements, we use these informal tests as a way to demonstrate the features and functionalities of the system in order to generate in-depth, timely, and specific feedback to the developer. Of course, any unusual behavior of the system will show as bugs or defects to be corrected for the next iteration of demonstration. Depending on the workload and available manpower, these live demos of partial working systems can happen on a weekly or monthly basis. Not all projects can enjoy the luxury of close participation with the enduser, and we are in a unique position to have the commitment from the researchers and staff members of the CCIR to participate in developing a comprehensive set of features in a baseline Imaging MIMI system that cannot be found in a single existing system.

The remaining three principles of web-interface, decentralized content management and employing the latest technology are common-sense: the web-interface provides uniform and wide accessibility; menu-driven interaction provides more control
over data input, output and presentation; decentralized content management reduces the overall management overhead after the system is deployed. However, achieving these requires a long-term vision and knowledge in several related fields. We are again in a unique position to follow these principles during development due to Case Western Reserve University’s distinct collaborative environment across management centers. Our guiding principles should be relevant and helpful for similar efforts.

3.2.2 System overview

The baseline MIMI consists of two main components: the Meta Server and the Data Server. The Meta Server is the common front-end for MIMI’s functionality. It is called “Meta Server” due to its role in managing all relevant alpha-numeric data: user profiles, project information, scheduling information, data storage address information, access control, etc. It supports a web-interface for data downloading after experimental data are acquired, using the client-server paradigm [56]. Administrative functionalities are also support by the Meta Server, such as validating user-supported information, assigning access privileges, and confirming requested scanning sessions. In a manager’s role, a user can launch the usage-statistics program to monitor resource usage and generate statements for fees for the core. The Meta Server is also involved in the final step of data-flow: after imaging data are acquired, a Java program can be launched from the scanner workstation (usually a PC) which receives input about the address of a local folder containing the acquired data and a RAID directory path on the Data Server representing the location where the data will be
stored. The RAID path consists of meta-data automatically generated by the Meta Server to represent the unique, humanly readable directory path on the Data Server.

The Data Server is the back-end for storage management of acquired imaging data. It uses a standard folder hierarchy for storage. To safeguard data from network viruses [76] and prevent unauthorized access, the Data Server operates behind a hardware firewall with communication permitted only with the Meta Server and with the local area network (LAN) PCs attached to scanners. The Meta Server and the Data Server together achieve common functionalities of a data warehouse [80].

3.2.3 Design and implementation

MIMI is designed to support a core facility’s administrative and scientific workflows in a single system. The administrative workflow includes managing profile data on users and research projects, scheduling scanning sessions, billing services, and compiling performance statistics to monitor resource usage. The scientific workflow consists of managing scientific data and disseminating them to the relevant researchers through a common web-interface.

Three data models will be used for the administrative workflow [Figures 3.1, 3.4, and 3.7]. The description of these data models follows the activity diagram specification of the Unified Modeling Language, or UML [26]. A solid dot represents the initial state. Rectangular boxes and round-corner boxes denote activities and objects, respectively. Solid arrows specify transitions between activities. Dashed arrows entering or leaving an object represent modification/creation or retrieval, respectively.
We begin by explaining how each data model supports the administrative workflow. We then describe how each data model is implemented using Plone. We conclude this section by showing how the scientific workflow is addressed through the data-flow model [Figure 3.11].

3.2.3.1 User profile model

![User profile model diagram]

Figure 3.1: The user profile model

The user profile model [Figure 3.1] specifies the behavior of the user management segment of the administrative workflow. The model is aimed to ease the data entry burden of a core facility by allowing the user to enter data, which will be validated by a manager in a core facility for it to become effective. The first action indicated by the model is searching for an existing user profile for a specific user. If the user profile does not exist, then it must be requested by the user as a pending user profile. Otherwise, if the user profile exists, then it will be displayed. The user profile model
then proceeds to define actions for a pending user profile. A pending user profile that does not meet the criteria for approval needs to be modified by the user or a core facility manager. A pending user profile that meets the criteria for approval can be approved by a core facility manager.

Figure 3.2: UML object diagram of Plone’s objects and their inheritance relationships

The Plone implementation of the profile model uses the profile object, which stores details about core facility users. The profile object resides at the top level of the Plone object hierarchy [Figure 3.2]. It captures information using the following string attributes: first name, last name, email address, institution, department, phone, fax, address, city, state, zip code, country, login ID, and status. The last two attributes store a user’s ID for logging into Plone and a value for pending (P) or approved (A) status, respectively. The profile object also contains a roles attribute that stores a list of user roles. Plone accesses the value of the roles attribute to determine a user’s access privileges. The four possible user roles are: Principal Investigator (PI), Co-Investigator (CI), Operator, and Manager (a user can assume multiple roles). Users with the PI role are researchers who have active research projects. Users with the CI
role are collaborators who work with other researchers on research projects. Operators represent users who are qualified to operate equipment. Managers are core facility staff members with “super-user” privileges, and they have access to all of MIMI’s functionalities.

Figure 3.3: Left: Profile request form. Right: Profile management interface, which is used to modify or approve pending profile objects.

When a user is granted the privilege to create a profile object, a profile request form is presented with input fields to capture information such as a user’s email address and phone number[Figure 3.3]. Once a user submits the profile request form, Plone creates a pending profile object with its status attribute set to “P.” Core facility staff
members with profile objects that contain “manager” as a value for the roles attribute are ultimately responsible for approving all pending profile objects through Plone’s profile management interface [Figure 3.3]. The main criterion for approval is verifying that a profile object’s login ID is associated with the right contact information such as a user’s email address and phone number. Approving profile objects guards against malicious users who attempt to pose as others to gain access to private information.

3.2.3.2 Project model

The project model [Figure 3.4] specifies the behavior of the project information management segment of the administrative workflow. The initial state of the model consists of a decision node that returns “Yes” or “No” depending on whether an existing project is selected. If “Yes,” then the information about the selected project is displayed. If “No,” then a user can request a (pending) new project. The project model then specifies actions for both pending and approved projects. A pending project that does not meet the criteria for approval must be modified by the user or a manager, while a pending project that meets the criteria for approval can be approved by a manager. An approved project can be modified by its owner to grant privileges for specified collaborators (among the existing users) to access the associated experimental data.

The project model is implemented in Plone using a project object which captures information about a specific research project. The project object uses the following attributes to capture the associated information: name, PI, CIs, IACUC number, grant number, account number, and description. The name attribute stores the title
of an active grant or a pilot study. The PI attribute, which stores the ID of a profile object that contains “PI” as a value for its roles attribute, links a project object with a principal investigator. The CIs attribute specifies a project’s collaborating users by storing the IDs of existing profile objects that contain “CI” as a value for their roles attribute.

A project request form [Figure 3.5] is implemented in Plone to allow a user to request a project object which, if approved, will have the user as its owner. This form contains input fields to capture project details as mentioned in the previous paragraph. It also presents a checkbox interface to allow a user to select user profile objects for inclusion as values for CIs. Once a user submits a project request form, Plone creates a pending project object with its status attribute set to “P.” Core facility staff members with profile objects that contain “manager” as a value for the
Figure 3.5: Top: Project request form. Bottom: Project management interface, which is used to modify or approve pending project objects.

roles attribute are ultimately responsible for approving all pending project objects through Plone [Figure 3.5]. The criteria for approval include checking that a project object’s grant number and account number are valid.

A user can also use Plone to view approved project objects that he/she is associated with as the project PI or a collaborator. In the case that new collaborators arrive, or old ones depart, a user may modify the list of collaborating users for these
Figure 3.6: The interface for viewing approved project objects and modifying their list of collaborating users

approved project objects through a web-based checkbox interface [Figure 3.6]. For security purposes, a user cannot create new users (profile objects) and may only select collaborators for a project from existing profile objects. Relegating the management of project collaborators to project owners (PIs) is an example of decentralized content management which alleviates the data management burden of a core facility. A collaborator of a project is typically granted the privilege to access experimental data resulting from the project.

3.2.3.3 Session model

The session model [Figure 3.7] specifies the behavior of the scheduling, billing, and usage-statistics compilation segments of the administrative workflow. The initial state of the model is a decision node to determine which actions to perform depending on whether completed or scheduled sessions are selected. If completed sessions are selected, then usage-statistics compilation can be performed. If scheduled sessions are selected, then a calendar with the scheduling information will be displayed. The session model specifies further actions for scheduled sessions and empty time slots
on the schedule. A scheduled session that is invalid (incorrectly scheduled) will be canceled, while a scheduled session that is valid (correctly scheduled) will be followed by input for billing details. An empty time slot on the schedule permits the scheduling of a new session within the selected time interval.

The Plone implementation of the session model uses the session object, which represents a scheduled or completed session for an imaging system. The session object resides at the lowest level of the Plone-object hierarchy. It stores information using the following attributes: imaging system name, date, time slot, project, operator, scanned items, time duration, total cost, billed (stores boolean value), status, notes, and Data Server Path (a.k.a., the RAID path). The project attribute stores the ID
of a project object related to the session object. From the project object, relevant project information such as a PI's name will be automatically retrieved and displayed on the session schedule. The operator attribute stores the ID of a profile object that has “operator” as a value for its roles attribute (the operator attribute tracks the user who operates an imaging system during a session). The scanned items attribute stores the IDs of entities (e.g., small animals, cell plates) that are used during a session, and the status attribute stores a value of scheduled (S) or completed (C). We will describe the Data Server Path Attribute in Section 3.2.3.4.
MIMI features a web-based scheduling interface for imaging systems [Figure 3.8]. The scheduling interface uses a combination of DHTML and AJAX to approximate the response-speed and the look-and-feel of a desktop application. Users can create a new session object by dragging the mouse cursor over an open time slot that spans at least one 30-minute interval and then selecting a research project object. A new session contains values for the following attributes: imaging system name, date, time slot, project, and status (S). A user can then use MIMI’s scheduling interface to perform cancellations or to access the supplemental billing form to choose values for the remaining attributes.

MIMI’s supplemental billing form [Figure 3.9] contains fields that capture billing details such as time duration and cost information. It also allows a user to select the profile object for inclusion into the operator attribute. A user can also use the supplemental billing form to select scanned items by choosing group objects. A group object represents a collection of entities that have similar characteristics. For
example, 20 female mice with the same vendor and strain can translate into a single
group object. A group object uses the following key attributes to store information:
name, species, strain, vendor, and item IDs. The item IDs attribute stores a list of
unique IDs for each item of a group object. A user who submits the supplemental
billing form initiates the process of automatic cost computation. MIMI then sets
a session object’s status attribute to completed (C) and updates the values of the
remaining attributes.

Figure 3.10: The interface for compiling resource usage summary statistics, with
example results

The resource usage compilation capability can allow a core facility to regularly
track the usage of its equipment and provide important and useful summary statis-
tics on different aspects of the daily operations of a core. MIMI can generate perform-
ance assessments of imaging systems with different time-intervals. When compiling
a performance assessment, Plone locates the relevant completed session objects and
sums the values of their time duration attributes. Plone’s built-in search interface
is modified to filter completed session objects using criteria such as PI, project, and date range, through text-fields and drop-down lists [Figure 3.10].

3.2.3.4 Data-flow

MIMI addresses a core facility’s scientific workflow with the data-flow process. Figure 3.11, in-tandem with Table 3.1, summarizes the data-flow process, which is explained in more detail in the remainder of the section.

Figure 3.11: The data-flow process, with steps described in Table 3.1

MIMI implements a data-flow process which seamlessly links imaging data with the associated session meta-data. With the completion of an imaging session, imaging data are stored in a standard folder hierarchy on the attached local workstation PC. The operator then double-clicks a jar file on the workstation PC. The jar file is a Java executable program for the Uploader application which is responsible for transferring the scanned imaging data to appropriate folders on the Data Server. After launching the program, the user looks up for the correct session object from the Meta Server and retrieves the value of its Data Server Path Attribute – a value automatically generated...
when a session object is created from the scheduling interface. The Data Server Path value is a string with six main parts (year, month, day, PI name, imaging system name, time slot) that uses the back-slash as a delimiter. Since MIMI automatically accounts for scheduling conflicts, the Data Server Path value represents a unique storage location on the Data Server. The operator copies the Data Server Path value, pastes it into a textbox of the Uploader application, and selects the local directory path for the folder containing the imaging data. Once the origin and destination for the imaging data are given, the Uploader application initiates a data transfer session with a single mouse-click.

<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td>User requests to view scheduled session.</td>
</tr>
<tr>
<td>02</td>
<td>Meta Server replies with session object.</td>
</tr>
<tr>
<td>03</td>
<td>User copies Data Server Path string into Uploader application.</td>
</tr>
<tr>
<td>04</td>
<td>Uploader application sends scientific data to Data Server.</td>
</tr>
<tr>
<td>05</td>
<td>Data Server stores scientific data.</td>
</tr>
<tr>
<td>06</td>
<td>Person requests to view downloading interface.</td>
</tr>
<tr>
<td>07</td>
<td>Meta Server replies with downloading interface.</td>
</tr>
<tr>
<td>08</td>
<td>User issues download request.</td>
</tr>
<tr>
<td>09</td>
<td>Meta Server forwards download request to Data Server.</td>
</tr>
<tr>
<td>10</td>
<td>Data Server sends scientific data to Meta Server.</td>
</tr>
<tr>
<td>11</td>
<td>Meta Server forwards scientific data to internet PC.</td>
</tr>
<tr>
<td>12</td>
<td>User stores scientific data on internet PC.</td>
</tr>
</tbody>
</table>

Table 3.1: The steps of the data-flow process

At the receiving end of the data transfer process, the Data Server runs a Receiver Script that listens continuously for requests from active Uploader applications. For all incoming requests, the Receiver Script first obtains the Data Server Path string. The script then fetches an incoming file’s path and its name and concatenates them to the Data Server Path string to form an absolute storage path. The Receiver Script
parses the absolute storage path into a valid folder hierarchy and creates any missing folders to form a unique storage location. The script then creates an empty file object and retrieves its contents by streaming binary data in 65,535 byte increments (the entire cycle repeats until all files transfer successfully to the Data Server).

Once imaging data are moved to the Data Server, they can be immediately downloaded by their owners and collaborators through the Meta Server. MIMI supports this step with a Retrieval Script that runs on the Data Server and listens continuously for requests by the Data Request Script that runs on the Meta Server. The communication process begins when the Data Request Script accesses a session object, obtains its value of the Data Server Path Attribute, and sends this value along with a relative folder path to the Retrieval Script. The Retrieval Script joins the Data Server Path value and the relative folder path to form a query path. The Retrieval Script then opens the query path on the Data Server and obtains a list of its files and folders (if any). The script then iterates through the list, computes file and folder sizes, and forwards these details to the Meta Server. The Meta Server dynamically constructs the visual downloading interface [Figure 3.12] and sends it to the user. After the user selects files or folders to download, the Data Request Script builds a list that holds their path strings and sends it to the Retrieval Script. The Retrieval Script creates a temporary zip file and populates it by iterating through folder paths in the list and fetching any files. In the end, the Retrieval Script sends the zip file to the Data Request Script, and the Data Request Script forwards it to user’s local desktop. In the case that a file is larger than 1 GB, the Retrieval Script virtually partitions the file and allows the user to download individual pieces. When a user encounters a
folder larger than 1 GB, it is also possible to download only a subset of its contents at one time.

![File List]

Figure 3.12: The interface for downloading scientific data, where a red “X” denotes a file or folder that has been selected for downloading.

An innovative feature of MIMI’s implementation of the data-flow process is the Data Server Path Attribute which enables the treatment of imaging data as binary files. This unleashes MIMI from the complexity and variety of image file formats (e.g., dcm, nifti, analyze) and avoids conversion to any standard data formats. The necessary meta-data, usually stored as header information, reside in the portable path names for the folder hierarchy. They also serve as imaging data’s unique IDs.

The baseline MIMI, which consists of approximately 100 files and 14,000 lines of code, has been deployed and is in full use at the Case Center for Imaging Research (CCIR). The CCIR is a state-of-the-art imaging core facility of Case Western Reserve University’s Medical Center and of the University Hospitals of Cleveland. It hosts multiple major imaging systems such as a Siemens MicroPET and a Bruker Biospec 9.4 Tesla MR.
The Meta Server and the Data Server are deployed with a carefully chosen set of hardware and software components. The Meta Server runs on a Dell PowerEdge with dual 3Ghz Intel Xeon processors, 4GB of DDR2 RAM, and two 300GB 10K RPM Ultra-SCSI hard drives. It operates using Redhat Linux and runs an Apache front-end for Secure Sockets Layer (SSL) transmission.

The Data Server operates under the Windows 2003 operating system and provides a redundant array of independent disks (RAID) with eight 300GB hard drives connected with Dynamic Network Factory’s 8-channel controller handling the RAID-5 functionality.

3.3 Results

In this section, we present MIMI’s usage analysis followed by a performance analysis. The usage analysis gives a profile of MIMI’s usage statistics over an 18 month period with respect to the number of users, imaging sessions, and scientific data uploads/downloads. The performance analysis elaborates on MIMI’s scientific data transfer rates and the presence of bottlenecks when carrying out specific steps of the scientific workflow.

3.3.1 Usage analysis

Using MIMI’s usage-statistics compilation capability, we performed a usage analysis of MIMI at the CCIR. Since its initial deployment in early 2006, MIMI has served approximately 150 principal investigators, collaborating investigators, and research
assistants. During this period, a total of approximately 1600 distinct sessions have been scheduled through MIMI, spanning an 18 month period or 400 working days. This translates to 4 scheduled sessions per working day. Among all sessions, about 31% are linked to scientific data. This entails that imaging data have been transferred to the Data Server using MIMI’s data-flow process at the frequency of just over one time per working day. Users also typically download the acquired data on the same day, so data downloading through the Meta Server occurs about one time per working day. This does not include data downloading activities by collaborators or repeated data downloading afterwards for various reasons. During the same period, MIMI cumulated 0.7 terabytes of fresh imaging data, which translates to a data acquisition rate of 1.75 gigabytes per working day.

The distribution of accrued content objects during the 18 month period is presented in [Table 3.2]. With respect to the anticipated capacity, the Meta Server should be able to handle over 1,000 registered users, 500 projects, 1,000 groups, and 10,000 sessions. The Data Server is designed to maintain 20 terabytes of online data.

<table>
<thead>
<tr>
<th>content type</th>
<th>sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>registered users</td>
<td>150</td>
</tr>
<tr>
<td>projects</td>
<td>125</td>
</tr>
<tr>
<td>groups</td>
<td>120</td>
</tr>
<tr>
<td>sessions</td>
<td>1600</td>
</tr>
<tr>
<td>acquired images</td>
<td>0.7TB</td>
</tr>
</tbody>
</table>

Table 3.2: MIMI’s content object and data statistics
3.3.2 Performance analysis

3.3.2.1 Transfer rates

We estimated the corresponding scientific data transfer rates when using MIMI to carry out the steps of the scientific workflow\(^1\). When carrying out each step with MIMI, we worked with three test cases to provide a broad sampling of scientific data. Each test case consisted of a baseline number of files that used one of three imaging systems, which are described in Section 1.1.1: The Siemens Medical Solutions MicroPET, the Gamma Medica X-SPECT, and the Bruker Biospec 9.4 Tesla MR. For the last two test cases, we also included the baseline number of files ± 25% to account for the standard deviation. For example, the number of files spanned 384, 512, and 640 for the Gamma Medica X-SPECT test case [Table 3.3]. As a whole, the three test cases represented the following file formats, which are associated with the MicroPET, X-SPECT, and 9.4 Tesla MR systems, respectively: raw list data, raw camera data and proprietary slice images, and DICOM images.

<table>
<thead>
<tr>
<th></th>
<th>MicroPET</th>
<th>X-SPECT</th>
<th>9.4 Tesla MR</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline number of files</td>
<td>1</td>
<td>512</td>
<td>900</td>
</tr>
<tr>
<td>baseline number of files x 0.75</td>
<td>n/a</td>
<td>384</td>
<td>675</td>
</tr>
<tr>
<td>baseline number of files x 1.25</td>
<td>n/a</td>
<td>640</td>
<td>1125</td>
</tr>
</tbody>
</table>

Table 3.3: File numbers which are associated with three imaging systems and used as test cases

\(^1\)The reference point for data transfer rates is 19.8 MB per second. This rate was achieved by using file-sharing to download scientific data from the Meta Server to a user’s local workstation PC on the CCIR network.
Figure 3.13: The CCIR’s network topology. Labels 1-5 are explained in Table 3.4.

*Uploading data from imaging systems’ workstation PCs to the Data Server.* The initial step that we analyzed was transferring scientific data from imaging systems’ local workstation PCs to the Data Server. Figure 3.13 shows the locations of both imaging systems’ workstation PCs and the Data Server in the CCIR’s network topology. From the results contained in Figure 3.14, we estimate that the average scientific data transfer rate from imaging systems’ local workstation PCs to the Data Server is approximately 22.6 MB per second, with a range from a low of about 2 MB per second to a high of 82 MB per second. We see that the low-range scientific data transfer rate occurs when MIMI handles a multitude of small files that are generated.
Figure 3.14: The transfer rates for uploading scientific data from imaging systems’ workstation PCs to the Data Server

by the Bruker Biospec 9.4 Tesla MR imaging system, while the high-range transfer rate involves one large scientific data file created by the Siemens Medical Solutions MicroPET. As a result, we infer that the transfer rate for uploading scientific data from imaging systems’ workstation PCs to the Data Server is inversely proportional to the number of files that are uploaded, with one possible explanation being that the

<table>
<thead>
<tr>
<th></th>
<th>An imaging system’s workstation PC</th>
</tr>
</thead>
<tbody>
<tr>
<td>01</td>
<td></td>
</tr>
<tr>
<td>02</td>
<td>The Data Server</td>
</tr>
<tr>
<td>03</td>
<td>The Meta Server</td>
</tr>
<tr>
<td>04</td>
<td>A user’s local workstation PC on the CCIR network</td>
</tr>
<tr>
<td>05</td>
<td>A user’s workstation PC on the CWRU network</td>
</tr>
</tbody>
</table>

Table 3.4: Key computers that are part of the CCIR’s network topology, which is illustrated in Figure 3.13
Figure 3.15: The transfer rates for copying scientific data from the Data Server to the Meta Server.

Data Server must devote processing overhead for creating multiple directories and file objects when dealing with multiple small files, which is not the case when the Data Server handles one large scientific data file. The resulting implication is that the transfer rate depends on the number of files.

**Copying data from the Data Server to the Meta Server.** We next analyzed the step of transferring scientific data from the Data Server to the Meta Server. Figure 3.13 shows that the Data Server and the Meta Server are connected together within the CCIR’s network topology without any intermediary PCs between them. As is evidenced by Figure 3.15, we find that transferring scientific data from the Data Server to the Meta Server occurs at an average transfer rate of about 31.3 MB per second across all three test cases. Although the number of scientific data files varies
Figure 3.16: The transfer rates for downloading scientific data from the Meta Server to a user’s local workstation PC on the CCIR network for each test case, the scientific data are always in the form of one file (a zip file or binary piece) when they travel from the Data Server to the Meta Server, which is a possible explanation for the lack of substantial range in transfer rates.

Download data from the Meta Server to users’ workstation PCs. Next, we analyzed the step of downloading scientific data from the Meta Server to users’ workstation PCs. We performed the analysis with two types of workstation PCs: a user’s local workstation PC on the CCIR network in addition to a user’s workstation PC on an external network, both of which utilized a gigabit ethernet connection. Figure 3.13 shows the locations of both the Meta Server and these two users’ workstation PCs in the corresponding network topology.

Based on the results from Figure 3.16, we estimate that the average scientific data
Figure 3.17: The transfer rates for downloading scientific data from the Meta Server to a user’s workstation PC on the CWRU network

We also estimated the scientific data transfer rate from the Meta Server to a user’s local workstation PC on the CCIR network is approximately 24 MB per second for all three test cases. Although a user’s workstation PC on the CWRU network is technically not local to the Meta Server, we see that the average scientific data transfer rate from the Meta Server to a user’s workstation PC on the CWRU network is about 28.7 MB per second, which is slightly higher than the average transfer rate from from the Meta Server to a user’s local workstation PC on the CCIR network. This is the case since the Meta Server uses two Internet Protocol (IP) addresses: a private IP address and a public CWRU
IP address. The private IP address is visible to users’ local workstation PCs on the CCIR network, while the public CWRU IP address is visible to CWRU network PCs, which treat the Meta Server as a native component of the CWRU network.

3.3.2.2 Bottlenecks

As a complement to the estimated transfer rates found in Section 3.3.2.1, we analyzed the bottlenecks that are present when using MIMI to carry out the steps of the scientific workflow.

We first analyzed the bottlenecks for uploading scientific data from an imaging system’s workstation PC to the Data Server. It is possible to improve the scientific data transfer rate from an imaging system’s workstation PC to the Data Server because the ceiling of the estimated scientific data transfer rates shown in Figure 3.14, which is approximately 82 MB per second, only saturates about 66% of the available network bandwidth of 1 Gigabit per second. The bottleneck which limits the transfer rate in this case is either a networking limitation or a hardware limitation. Network latency, which is the time delay incurred when a data packet moves from one PC to another, can affect scientific data transfer rates. However, it is not significant when uploading data from an imaging system’s workstation PC to the Data Server since an imaging system’s workstation PC has a direct local network path to the Data Server. Scientific data that travel from an imaging system’s workstation PC to the Data Server do not have to “hop” through other PCs, which results in negligible network latency of < 1 millisecond that was estimated with the execution of a tracert command [Table 3.5] [4]. We thus conclude that the rate at which an imaging sys-
tem’s workstation PC reads scientific data from its hard disk is the bottleneck when uploading scientific data to the Data Server. The Data Server is optimized to store scientific data quickly using its RAID but must wait on the data to be retrieved from the imaging system’s workstation PC.

Table 3.5: The results of executing the `tracert` command from an imaging system’s workstation PC to measure the number of hops and network latency to the Data Server

We next analyzed the bottlenecks that occur when transferring scientific data from the Data Server to the Meta Server. Since the Meta Server is similar to an imaging system’s workstation PC in hardware specifications, we deduce that the rate at which the Meta Server stores scientific data to its hard disk is the bottleneck when transferring scientific data from the Data Server.

Table 3.6: The results of executing the `tracert` command from a user’s workstation PC on the CWRU network to measure the number of hops and network latency to the Meta Server

Although downloading scientific data from the Meta Server to users’ local CCIR or CWRU network PCs occurs at average transfer rates of approximately 24 and 28.7 MB per second as is evidenced by Figures 3.16-3.17, these rates are the result of
both a network limitation and computational overhead. Downloading scientific data from the Meta Server to users’ local CCIR or CWRU network PCs entails multiple data “hops” between other PCs, which leads to a tangible network latency that was measured to be about 3 milliseconds with the execution of a tracert command [Table 3.6]. In addition, computational overhead results from decrypting scientific data on a user’s local CCIR or CWRU network PC after they are originally encrypted on the Meta Server.

3.4 Discussion

Based on our deployment experiences at the CCIR, we find the baseline MIMI to be a practical solution for managing an imaging core facility’s scientific and administrative tasks. Administratively, a core facility can manage users and research projects, schedule its equipment, perform billing, and compile statistical performance assessments. Scientifically, a core facility can manage scientific data and disseminate them to researchers using the data-flow process. The data-flow process has been the main distinction and focus of effort due to the presence of heterogeneous data, whose integration can facilitate the discovery of new scientific knowledge. MIMI uses a “universal data format” which is compatible with all types of scientific data to enable storage and retrieval that are format and system independent, respectively. Due to the generality of design considerations, MIMI can be used for other kinds of core facilities, such as proteomics and flow cytometry.

The results of estimating the scientific data transfer rates when using MIMI show
that the transfer rate for downloading scientific data from the Meta Server to a user’s workstation PC does not correlate with the number of files. However, the transfer rate for uploading scientific data from an imaging system’s workstation PC to the Data Server increases as the number of files decreases. As a result, placing multiple files inside a zip file and then uploading them to the Data Server is an approach whose computational overhead for creating the zip file may be offset by the simplicity of uploading one file versus multiple files. Of course, the transfer rates were estimated from using MIMI during a typical workday, which represents the worst-case scenario for evaluation: the “rush-hour” of MIMI usage. We thus conclude that the estimated transfer rates are floor values, which we anticipate will increase with the formulation of an off-hour transfer strategy.

In addition, we conclude that in order to combat the bottlenecks that are present when using MIMI to carry out the steps of the scientific workflow, three aspects can be improved: The rate at which an imaging system’s workstation PC reads scientific data from its hard disk, the rate at which the Meta Server stores scientific data to its hard disk, and the computational capabilities of local CCIR and CWRU network PCs. Aspects one and two can be improved with the introduction of faster hard disk drives or a RAID, while the last aspect can be improved with the use of a faster processor or multiple processors.
Chapter 4

The Proteomics MIMI

4.1 Introduction

Although the Imaging MIMI is fully deployed at the Case Center for Imaging Research (see Chapter 3), our general approach is intended to be applicable to other domains such as proteomics. Both proteomics and imaging core facilities share the common entities which are described in Chapter 2. However, the proteomics domain differs from the imaging domain in two aspects: types of research materials and experimental workflow. While the types of research materials used in imaging include small animals such as rats and woodchucks, proteomics uses research materials such as tissue samples and liquid solutions (which are smaller, more portable, and easier to handle) when generating scientific data that include annotated two-dimensional gels and mass spectrums [93]. In addition, in imaging, a session, or study, consists of only one step, which involves carrying out a scanning experiment using one imaging system, where for example, a researcher obtains images of a rat using a Siemens
MicroPET system. However, in proteomics, a typical study is comprised of multiple steps [28]. A proteomics researcher who performs a study uses multiple systems before obtaining the desired results. For example, a proteomics researcher completes studies such as two-dimensional gel processing and liquid chromatographic separation combined with mass spectrometry [5] only with the use of systems in an atomic fashion. Performing a proteomics study is equivalent to an active assembly-line process that does not output scientific data until the end is reached. The sequence of steps necessary to carry out a proteomics study is known as the experimental workflow. As a byproduct of the experimental workflow, proteomics core facilities are faced with significant challenges, which include:

1. experimental workflow management, i.e., defining the steps of a study in a manner that tailors to both the researcher and the staff members of a proteomics core facility

2. optimized scheduling of multiple resources for multi-step studies and of one resource for simultaneous studies at the same time

3. optimized scheduling of personnel who are responsible for operating resources

4. effective and reliable tracking of research materials

5. efficient creation of custom statements and invoices for multi-step studies

There is evidence in the literature of systems that facilitate experimental workflow management. For example, Goodman et al. [33] described the LabBase system for genomic laboratories, while both Medeiros et al. [64] and Vossen et al. [97] elaborated
on a Workflow-based Architecture to support Scientific Applications (WASA), which used commercial software as an interface for experimental workflow management. In addition, Lee et al. [54] introduced a system for managing proteomics experiments, which was also the purpose of Siemens’ MIPortal [81]. However, there are currently no software solutions that can comprehensively address the challenges described above, including experimental workflow management. This chapter describes the relevant developmental and implementation details of the proteomics MIMI, which is a system that addresses these challenges and is deployed at a proteomics core facility. In Section 4.2, we detail what developmental environment was chosen for development and introduce a high-level data model of proteomics MIMI. In Section 4.3, we elaborate on relevant design and implementation aspects, which are contrasted to the Imaging MIMI system deployed at the Case Center for Imaging Research (see Chapter 3). In Section 4.4, we provide concluding remarks.

### 4.2 Development

The proteomics MIMI is deployed at the Case Center for Proteomics, or CCP. The proteomics MIMI has been the culmination of dedicated work by Adam D. Troy (adam.troy@cwru.edu) that spans the beginning of September 2006 to the conclusion of April 2007. The development cycle for the proteomics MIMI lasted only about one-half of a year because the Imaging MIMI was used as point of reference, and development did not have to start completely anew. Although it was not sufficient to merely transplant the existing Imaging MIMI from the CCIR to the locale of the CCP,
the feature-set of the Imaging MIMI was expanded to tailor to the functionalities that addressed the CCP’s informatics challenges.

*Developmental environment.* As with the Imaging MIMI, the developmental environment of choice for the proteomics MIMI is open-source. Once again, using an open-source developmental environment leads to low developmental cost and is conducive to a system’s expandability and modifiability. The proteomics MIMI’s main developmental environment is called the Google Web Toolkit. The Google Web Toolkit was chosen for its web-based interface for development and its built-in feature-set for facilitating the development of dynamic web applications. The Google Web Toolkit’s Java-based front-end allows the developer to create a program that is automatically translated into web-friendly Hypertext Markup Language (HTML) and Javascript code. The translation process from Java code to Javascript and HTML code is attractive for cross-web-browser compatibility, since the Google Web Toolkit does not burden the developer with creating multiple versions of HTML and Javascript code for different web-browsers. Also, the Google Web Toolkit’s extensive generation of Javascript code promotes the use of client-side interactions and AJAX, both of which decrease server-load. The Google Web Toolkit distribution is even available for multiple operating systems such as Mac OS, Windows, and Linux, so a preferred environment for development can be selected.

*High-level data model.* Figure 4.1 displays the proteomics MIMI’s high-level data model that was used during development. This data model, which follows the specifications of the Unified Modeling Language (UML), was created during the early stages
of development. We show this data model to emphasize its complexity with respect to the Imaging MIMI’s data model, which is found in Section 2.3. Developing the proteomics MIMI entailed defining objects and relationships that were more intricate in comparison to the objects and relationships defined when developing the Imaging MIMI. Two relevant objects that differentiate the proteomics MIMI from the Imaging MIMI are the study and step objects, which are discussed in Section 4.3.1.

4.3 Design and implementation

The proteomics MIMI is designed to address the informatics challenges of a proteomics core facility, which include supporting the experimental workflow. Managing the experimental workflow includes defining the steps for a study and checking that the
defined steps are valid. The proteomics MIMI also addresses the remaining challenges found in Section 4.1, which are more detailed instances of the challenges described in Chapter 1.

In the sections below, we outline the proteomics MIMI’s design (where relevant) and implementation details that explain how the experimental workflow is supported and how the informatics challenges of a proteomics core facility are addressed. In order to receive a valuable perspective on the intricacy behind the functionalities of the proteomics MIMI, we also provide a contrast with the Imaging MIMI by discussing how the proteomics MIMI expands upon the Imaging MIMI’s functionalities. In Section 4.3.1, we elaborate on supporting the experimental workflow. In Section 4.3.2, we discuss addressing scheduling needs, which include optimized scheduling of: multiple resources for multi-step studies, one resource for simultaneous studies at the same time, and personnel who are responsible for operating resources. In Section 4.3.3, we describe how research materials are tracked. In Section 4.3.4, we detail how statements and invoices are created.

4.3.1 Supporting the experimental workflow

The Imaging MIMI does not support experimental workflow management since imaging studies are typically one step, which eliminates the need to define custom multi-step studies. However, the proteomics MIMI uses the study template model for managing the experimental workflow [Figure 4.2]. The description of the study template model follows the *activity diagram specification* of the UML. A solid dot represents
the initial state. Rectangular boxes and round-corner boxes denote activities and objects, respectively. Solid arrows specify transitions between activities. Diamonds are decision nodes, which have outgoing solid arrows that are traversed only if the specified conditions in brackets are met. Dashed arrows entering or leaving an object represent modification/creation or retrieval, respectively. We next explain how the study template model supports the experimental workflow and then describe how the study template model is implemented using the Google Web Toolkit. We conclude this section by describing a use-case scenario for the study template model.

Figure 4.2: The study template model

The study template model specifies the behavior of the experimental workflow. The model’s goal is to ease the data entry burden of a proteomics core facility by allowing the user to enter data, which will be validated by a manager at the core facility before becoming effective. The initial state indicated by the model is a decision
node that returns “Yes” or “No” depending on whether an existing study template is selected. If “Yes,” then the selected study template is displayed. If “No,” then a user can request a (pending) new study template. The study template model then denotes actions for both pending and approved study templates. A pending study template with invalid steps that does not meet the criteria for approval must be modified by the user or a manager, while a pending study template that defines valid steps can be approved by a manager.

<table>
<thead>
<tr>
<th>object type</th>
<th>attributes</th>
</tr>
</thead>
</table>
| Study       | - unique identifier  
             |   - name  
             |   - details  
             |   - project ID  
             |   - status  
             |   - last activity  
             |   - invoice ID |
| Step        | - details  
             |   - start time  
             |   - end time  
             |   - actual duration  
             |   - study ID  
             |   - operator ID  
             |   - instrument ID  
             |   - service ID |

Table 4.1: The attributes of the study and step objects

The study template model is implemented with the Google Web Toolkit using study and step objects, which capture information about proteomics studies and their corresponding steps and contain various attributes [Table 4.1]. Each study object includes a unique identifier attribute, a project ID attribute, and a status attribute. The unique identifier attribute’s value, which is a string, can be accessed
for fast object retrieval. The project ID attribute, which stores a string value, links a study to an existing project. The status attribute contains either “P” or “A” entries for pending or approved, respectively. In addition, each step object contains a study ID attribute, which stores a unique identifier value and links the step object to a study object.

Figure 4.3: The experimental workflow management interface for viewing or modifying existing study templates

The experimental workflow management interface [Figure 4.3], which is implemented using the Google Web Toolkit, allows core facility managers to view or modify existing study templates. The background in Figure 4.3 contains the existing study templates that a manager can view, while the foreground displays an interactive Javascript interface that manages the selection, deletion, and ordering of the steps of a study template.
Figure 4.4: A use-case scenario for the study template model

*Use-case scenario.* Figure 4.4 displays a diagram of a use-case scenario for the study template model, which follows the *use case specification* of the UML [51]. Stick figures represent personnel, or actors. Horizontal ellipses denote a sequence of actions that are affiliated with at least one actor, and lines specify affiliations.

The first step in the use-case scenario occurs when a user requests an incorrect study through a study template which contains at least one misplaced or unrelated step. Both the user and a manager then view the incorrect study, but only the manager proceeds to modify its steps, which leads to subsequent study approval. The user and manager then access the approved study for reference purposes, which completes the use-case scenario.
4.3.2 Optimizing scheduling

The Imaging MIMI’s scheduling interface does not tailor to the needs of the proteomics domain since it does not support optimized scheduling of: multiple resources for multi-step studies, one resource for simultaneous studies at the same time, and personnel who are responsible for operating resources. The Imaging MIMI’s scheduling interface allows a user to schedule only a single-step study using one resource. Of course, a multi-step study can be divided into single-step studies, which can be scheduled individually through the Imaging MIMI’s scheduling interface. However, scheduling multiple single-step studies in sequence using the Imaging MIMI’s scheduling interface is inefficient because it requires a significant amount of work by the user and does not recognize the single-step studies as one multi-step study. The proteomics MIMI’s scheduling interface [Figure 4.5] differs from the Imaging MIMI’s scheduling interface since it supports all of the scheduling challenges faced by a proteomics core facility. We now describe how the proteomics MIMI’s scheduling interface supports each scheduling challenge.

Scheduling multiple resources for multi-step studies. The scheduling interface of the proteomics MIMI processes each individual step of a multi-step study across multiple resources in a cascading fashion. It starts from step one and automatically chooses an available time slot and then proceeds through the remaining steps. Dynamic programming, which is a recursive method for solving sequential decision problems [78], is the basis for the scheduling algorithm utilized by the scheduling interface [Table 4.2]:
Figure 4.5: The proteomics MIMI’s scheduling interface from a manager’s viewpoint, which is implemented using the Google Web Toolkit

(1) The scheduling interface obtains a study template with the steps of a study and treats these steps as sub-problems.

(2) It then uses a “bottom-up” programming approach and solves the sub-problems by choosing an available time slot for each step in a sequential fashion and under the constraint that the time slot must occur after the previous sub-problem’s occupied time slot, if any.

1. Foreach step of study template DO
2. IF previous step exists THEN
3. select time slot that occurs after previous step’s time slot
4. END IF
5. IF step is initial one of study template THEN
6. select earliest time slot
7. END IF
8. END FOR

Table 4.2: The scheduling algorithm, which is based on dynamic programming

For example, a proteomics user who wants to schedule a study that involves two steps, zip-tip cleaning and using the DECA-XP system, lets the scheduling interface
occupy a time slot for the first step and subsequently fill a time slot for the second step. The scheduling interface makes an educated guess for the amount of time to occupy for each step. We see that this automatic scheduling occurs with no manual intervention and is a substantial time-saver for a proteomics core facility.

* Scheduling one resource for simultaneous studies at the same time. A proteomics resource may have an input tray that holds multiple batches of research materials. Samples from different studies can be placed in the tray simultaneously. For example, a proteomics staff member may combine ten samples from study “A” with fifteen samples from study “B” in an input tray with twenty-five slots. The proteomics MIMI’s scheduling interface allows a user to input the exact number of research materials for each step of a study using a text-box interface. As a result, when the proteomics MIMI processes a study template and schedules a resource, it allocates the desired number of research material spots on the input tray, where applicable.

* Scheduling personnel who are responsible for operating resources. The proteomics MIMI’s scheduling interface allows a manager to use drop-down lists to select a person to operate a resource for each step of a study. Once a person is selected for a step, the scheduling interface then determines if conflicts exist by searching for the person’s unique ID in other scheduled steps.

### 4.3.3 Tracking research materials

The Imaging MIMI also falls short in providing a fully effective interface for tracking research materials during a proteomics study. Although the Imaging MIMI allows
researchers to manually specify the research materials (e.g., small animals) that apply to a single-step study, it does not have the ability to automatically interface with an external repository to transparently retrieve information about research materials. Having the ability to access an external repository such as a bar-code database is a significant requirement for a proteomics core facility since such a core facility tracks research materials that are small in size and easily susceptible to being lost or misplaced. In addition, a proteomics core facility that transitions to using a bar-code database for tracking research materials also experiences the following benefits:

(1) Reduction in data management overhead since research materials to not have to be entered manually multiple times.

(2) Reduction in user errors because typing mistakes are eliminated.

The proteomics MIMI can interface with an external bar-code database to retrieve information about research materials through MySQL queries, which allows a proteomics core facility to reap the latter benefits.

4.3.4 Creating statements and invoices

The Imaging MIMI automatically computes the cost of a single-step study for a statement by multiplying a default hourly rate with the study’s total elapsed time. However, this method of cost computation is insufficient for proteomics core facilities, since a proteomics core facility generally references the number of research materials used during each step of a study when performing the total cost computation for the study’s statement. The proteomics MIMI expands on the functionalities of the
Figure 4.6: A sample invoice for a study in Portable Document Format

Imaging MIMI by introducing three options for the cost computation for statements: fixed, per sample, and tiered sample. The fixed option allows a manager to enter the cost of a step of a study into a text-box interface. The per sample option uses two text-box interfaces, which capture the number of samples, or research materials, used for a step of a study in addition to the cost of each sample (the proteomics MIMI multiplies the values obtained from the two text-box interfaces to calculate the cost of the step of a study). The tiered sample option features three text-box interfaces: the first interface captures the number of samples used for a step, the second one captures the cost of the initial \( n \) samples used, where \( n \geq 1 \), and the third interface captures the cost of the remaining samples used for a step. To create an invoice for a multi-step study, the proteomics MIMI first calculates the costs of the steps of a study, sums them, and then integrates them and a detailed cost distribution into a Portable Document Format, or PDF, file, which is downloadable through a web-based interface [Figure 4.6].
4.4 Discussion

The proteomics MIMI addresses the informatics challenges of a proteomics core facility, with experimental workflow management being the main focus area based on the information that can be extracted. For example, knowing which experimental workflows provided useful results or should not be repeated can ultimately lead to more effective scientific research. We conclude that the proteomics MIMI’s implementation of the study template model using the Google Web Toolkit serves as a capable solution for managing the experimental workflow. The implementation of the study template model also introduces the study template, which allows researchers to define multi-step studies while still maintaining compatibility with single-step studies: a single-step study translates into a study template with one step. Also, the proteomics MIMI’s scheduling interface acts as a facilitator for a proteomics core facility’s scheduling needs, since manual processing of multi-step studies is kept to a minimum. In addition, the proteomics MIMI’s ability to interface with an external database to retrieve information on research materials eliminates redundant data entry for studies. Creating statements and invoices using the proteomics MIMI replaces the manual work required of core facility managers and also allows researchers to reference their entire cost history from a central location. Proteomics MIMI represents an informatics infrastructure that not only meets the informatics challenges of a proteomics core facility, but also provides a solution for the needs of a general core facility.
Chapter 5

Extreme Programming: A Case Study

5.1 Introduction

In response to the influx of computer-based applications in the late 1970s and early 1980s, software development headed towards a direction of identifying specific developmental stages and following them in a sequential fashion, which was thought to improve software management and provide a sense of discipline [9]. Such approaches to software development were known as the Systems Development Life Cycle (SDLC), or waterfall-based traditional developmental methodologies. However, these methodologies had limitations [10] that led to the appearance of alternative approaches, one of which was Extreme Programming, or XP.

XP is an agile software developmental methodology [62, 69]. It departs from the practices that are common to traditional developmental methodologies under the
waterfall designation. Unlike the traditional developmental methodologies, XP does not require that developers maintain up-to-date design documentation. The rationale behind eliminating the design documentation is that developers should focus on what they do best: the coding. The only mandatory forms of documentation defined by XP are the “user stories,” which represent desired functionalities from the customer’s point of view. Without formal documentation, XP emphasizes the code as the ideal means for gathering an in-depth understanding of a software project’s design intricacies. Also, with traditional developmental methodologies, there exists one developmental cycle that spans three steps: specification, implementation, and evaluation. However, XP takes a more fine-grained approach and divides a developmental cycle into multiple cycles that are smaller in scope. With XP, the developer is not required to implement a complete system in one sitting. Instead, the customer receives multiple software revisions for evaluation and witnesses development as an iterative process [Figure 5.1]. A byproduct of XP’s iterative developmental process is frequent communication with the customer. In contrast to waterfall-based developmental methodologies, XP does not relegate customer input as the final step of development. Instead, the customer is in a position to provide input at the end of each small developmental cycle.

Since the beginning of the twenty-first century, agile developmental methodologies, including XP, have taken the world of software development by storm [1]. With XP being widely recognized and representing an alternative to various waterfall developmental methodologies, case studies that evaluate XP’s viability through a set of distinct experiences have appeared (such case studies are useful because they typi-
Figure 5.1: XP’s high-level relationship between the customer and the developer
cally exhibit factors such as scale and unpredictability that are not common to staged experiments [75]). Muller and Tichy [68], Wilson [102], Keefe and Dick [46], and Noll and Atkinson [72] described their experiences when using XP in four separate educational settings. In addition, Knublauch et al. [50] provided details about following XP to implement agents for the flow of clinical information, while Karlstrom [45], Beck [11], and Layman et al. [53] detailed their experiences when applying XP for software projects in industry. These case studies were not performed on an empirical level, but they still offered useful perspectives on XP, with the general consensus being that, although adapting to XP posed multiple challenges, it was feasible to use XP as a developmental methodology for certain design projects.

Complex design projects with emerging, dynamic feature sets stand to benefit the most from using XP, since the relationship between the customer and the developers is a crucial component for successful development on a large-scale [47]. This chapter supports this assertion by describing how the development of an integrated informatics infrastructure for core facilities known as MIMI translates into specific practices of XP, i.e., while neither consciously nor strictly adhering to the XP, MIMI’s developmental methodology is a natural fit to many XP practices. In Section 5.1.1, we introduce and
summarize some relevant practices of XP. In Section 5.2, we explain how and when MIMI’s developmental methodology was carried out. We also describe how MIMI’s developmental methodology translates into using five different practices of XP. In Section 5.3, we offer concluding remarks.

5.1.1 The practices of extreme programming

When using XP, it is recommended that the developer follows twelve distinct practices. These practices, as a whole, define XP and adhere to its four core values of communication, simplicity, feedback, and courage [69]. We will now introduce and summarize the five practices of XP that are relevant to the results of this chapter.

The planning game. The planning game represents the initial step when using XP for development. It is a realization of two aspects: the needs of the customer and the capabilities of the developer. The customer provides the developer with “user stories” that summarize the desired functionalities. Both the developer and customer then work in-tandem to determine an acceptable schedule for the order of implementation. They arrange the “user stories” in a manner that best facilitates development.

Small releases. The practice of small releases divides a software design project into smaller, more developer-friendly sub-projects. Each sub-project, when fully implemented, represents a working addition to the final software product. With each small release, the developer moves one step closer to completing the entire software design project on a steady pace, and the customer receives an up-to-date software product for evaluation in a timely fashion. The rationale behind the practice of small
releases is captured well by the following proverb: “Do not bite off more than you can chew.”

\textit{System metaphor.} The system metaphor allows both the developer and customer to receive a common understanding of a system. It captures the gist of a system with terminology that is easy to comprehend. The system metaphor must not necessarily be at the system-level per se and can be defined on other levels as well. For example, on the high-level, the system metaphor can concisely express “what is currently being done,” and on the low-level, it can detail “how a part of the system works.”

\textit{Simple design.} The simple design practice entails that the developer implements only the bare necessities. The current requirements for implementation are sufficient for what the customer wants and nothing more. The simple design practice detracts the developer from implementing features that do not adhere to the customer’s needs for the current software release. The rationale behind using the simple design practice is sparing the developer from spending time on brainstorming and implementing functionalities that may never be used. A system may change in the future, but it is counter-productive to try to anticipate the changes.

\textit{On-site customer.} An on-site customer ties the developer to the customer during the entire developmental process. In the case that questions arise or requirements need to be changed, an on-site customer renders communication latency negligible, and the developer can quickly contact the customer.
5.2 The case study

We are qualified to report on MIMI’s development since we are the actual developers. We staged meetings with the customer from August 2005 to January 2006. These meetings occurred on a regular basis and provided the setting for outlining the initial set of design specifications. We obtained software requirements from the customer via either paper records, electronic files, or word-of-mouth. We moved to the customer’s facility during the first quarter of 2006 and continued development for approximately twelve months.

In the sections below, we document how MIMI’s development relates to the five practices of extreme programming that are described in Section 5.1.1. We also describe our developmental experiences, so that software developers who plan to follow the twelve practices of extreme programming may use these experiences as a learning tool for what to expect during some parts of development.

5.2.1 Planning the initial design

Obtaining MIMI’s initial set of design specifications corresponded to XP’s planning game practice. The customer provided design specifications which outlined MIMI’s functionalities and were partitioned in a nature that was similar to “user stories” [Figure 5.2]. The design specifications were useful because they allowed us to compartmentalize MIMI’s functionalities. We could easily inquire about a specific design feature without prior knowledge about the overall complexity of MIMI. MIMI’s design specifications were also an excellent starting point to the developmental process.
because they were divided into concise segments, which made them easy to work with.

The design specifications were built around the key common entities of a biomedical core facility. The customer provided the required design specifications with respect to an object model for the key common entities of a biomedical core facility (see Section 2.3). Each object from the object model associated with specific use-case scenarios that encapsulated the required functionalities. For example, the objects that represented people mapped to functionalities such as requesting a new profile or editing an existing profile.

Figure 5.2: Examples of MIMI’s design specifications, which were analogous to “user stories” from XP’s planning game

| - users can register themselves into the system  |
| - users can view their credentials            |
| - managers can modify users’ credentials     |
| - managers can look at credentials of all registered users |

| - principal investigators can report new research projects |
| - principal investigators can view their current research projects |
| - principal investigators can modify collaborators on research projects |
| - managers can modify principal investigators’ research projects |
| - managers can look at research projects of all principal investigators |

| - users can register research materials into the system |
| - users can view their research materials |

| - managers and users can schedule sessions |
| - managers can generate invoices for all scheduled sessions |
| - managers can compile performance statistics for completed sessions |
| - managers can modify details of completed sessions |
| - users can download scientific data from completed sessions |

The next step involved deciding on an effective order of implementation for each of MIMI’s functionalities. This was one step where MIMI’s developmental methodology separated from having similarities to XP’s planning game. The typical extreme programming planning game includes actively involving the customer with the devel-
oper to determine the best implementation order. In the case of MIMI’s development, the customer did not have a preference on the order for implementing each functionality and only emphasized wanting a fully-functional system as quickly as possible. Although the customer did not participate in the determining the implementation order, the initial set of design specifications still helped this process by providing well-defined “breaking points” for each of MIMI’s functionalities. We felt that working with pre-defined “breaking points” was much faster than arbitrarily dividing a large software project into custom design partitions and arranging them into an implementation order. We next describe how the preferred order of implementation for MIMI’s functionalities was determined.

Before even considering an implementation order for MIMI’s functionalities on an object-level, we needed to map out an implementation order for the objects. We found that mapping all functionalities to the defined objects from the object model resulted in an implicit order of implementation. More specifically, the object model contained associations between objects (see Section 2.3), e.g., a research project object mapped to a principal investigator object, and we discovered that the associations led to interdependencies on an implementation level as well. Hence, we identified the objects sans dependent associations – the people objects – and selected implementing their corresponding MIMI functionalities as the initial step in the developmental process. In turn, MIMI’s development shifted to formulating a developmental schedule for the remaining three objects and their applicable functionalities of MIMI. We decided to implement the remaining objects and their functionalities in the following order that respected the object associations found in the object model: research projects,
research materials, and sessions. Once the high-level order of implementation was
determined, we transitioned to planning the implementation order for each specific
object’s functionalities. In a similar fashion to determining the high-level order,
we ascertained use-case dependencies on a per-object basis (e.g., creating a session
object using a scheduling interface logically occurs before editing a session object)
and finalized implementation orders for MIMI’s functionalities that respected these
dependencies [Table 5.1].

<table>
<thead>
<tr>
<th>Functionality</th>
<th>People</th>
<th>Projects</th>
<th>Materials</th>
<th>Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>create or schedule</td>
<td>01</td>
<td>04</td>
<td>07</td>
<td>10</td>
</tr>
<tr>
<td>view</td>
<td>02</td>
<td>05</td>
<td>08</td>
<td>12</td>
</tr>
<tr>
<td>edit</td>
<td>03</td>
<td>06</td>
<td>09</td>
<td>13</td>
</tr>
<tr>
<td>upload and download data</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>16</td>
</tr>
<tr>
<td>manage billing and accounting</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>14</td>
</tr>
<tr>
<td>record session details</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>11</td>
</tr>
<tr>
<td>generate performance reports</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 5.1: The implementation order for MIMI’s functionalities by object

5.2.2 Performing the implementation

We released multiple software iterations of MIMI throughout the developmental pro-
cess, which was similar to XP’s practice of small releases. Each software iteration
incorporated a new MIMI feature in its entirety. We used the implementation orders
declared in Section 5.2.1 as a blueprint and treated each MIMI functionality associated
with an object as one complete feature. For example, we considered the functionality
of creating people objects as one cohesive feature. We noted that this feature, along
with others, could be broken down further into separate pieces. However, we treated
all such features of MIMI as atomic entities to maintain consistency.

We discovered that issuing multiple iterations of MIMI was a good implementation strategy for the most part. Each MIMI iteration served as a recurring motivator for completing a feature. We found that having to continually create software iterations provided distinct points of focus. Also, the customer feedback that followed each subsequent software iteration of MIMI helped immensely since we were able to fine-tune each software iteration before transitioning to developing the next one. The customer was not pressured to provide comprehensive feedback on a system-wide level. Releasing software iterations allowed us to process feedback on a MIMI feature basis, which was a more practical approach from our standpoint than dealing with feedback on a system-wide level.

We also found that issuing MIMI software iterations was not the perfect implementation strategy. Imposing self-defined implementation deadlines had the unintended consequence of serving as a problem source. We had the mindset to finish each software iteration as quickly as possible. However, developing two specific MIMI software iterations took more time than expected. What follows is a summary of the implementation steps that we performed when completing these two software iterations during MIMI’s development.

*Developing MIMI’s session scheduling interface.* As was evidenced by customer feedback, the initial version of the scheduling interface was not conducive towards an optimal user experience. A user who utilized the initial version had to select a project before being able to browse resource schedules and perform reservations,
which proved to be cumbersome for scheduling under multiple projects. The initial version of MIMI’s scheduling interface also featured drop-down lists for selecting the equipment, date, and time when scheduling a session, which required multiple mouse clicks and were prone to user errors. We revamped the initial scheduling interface by replacing the drop-down lists. To schedule a session with the revamped interface, a user hovered the mouse cursor over a specific date on the schedule of a resource and in turn, a window that contained a list of times appeared. The user then clicked a start time and an end time to schedule a session, which was faster than using the initial version of the scheduling interface. Of course, additional customer feedback dictated further modifications, which were reflected in a third and final version. The final version of the scheduling interface transplanted selecting a project from the first step to the last one when scheduling a session. The final version also replaced the appearance of the window that contained a list of times with fields whose values were pre-selected upon “dragging” an open time slot on the resource schedule.

*Developing MIMI’s scientific data downloading interface.* The initial version of the downloading interface did not account for the download size limit (2 GB file size) of certain web-browsers. Also, a user selected what to download with the initial version by clicking individual checkboxes that corresponded to specific files or folders, which became time-consuming for multiple selections. The initial version of MIMI’s downloading interface also initiated a server request when the user wanted to select a complete set of files and folders to download with a single mouse click. We modified the initial version of the downloading interface to work around the download size
limit of certain web-browsers by incorporating a new “sliding window” downloading mechanism. The downloading interface only displayed checkboxes corresponding to files and folders that, when clicked, would maintain the total download size below a pre-defined threshold. However, keeping the download size below a pre-defined threshold resulted in needing to issue separate download requests to retrieve a complete set of files and folders, which became burdensome to the user. After noting the customer’s feedback, we further revised the initial version of the downloading interface, which improved the user experience through client-side scripting by using Javascript to select or download files and folders without redundant server requests.

5.2.3 Using metaphors

When developing MIMI, we utilized two metaphors: high-level and low-level, both of which translated to XP’s system metaphor practice.

High-level metaphor. The high-level metaphor represented the entire MIMI system with a group of three unique workflows: administration, business tasks, and scientific data management. Each workflow consisted of MIMI functionalities that were logically related [Table 5.2]. The three workflows in-tandem worked well as informal milestones throughout the development process. When the deadline for a MIMI software iteration coincided with the final functionality associated with a workflow, development was tightly-focused. Implementing all of the MIMI functionalities of a workflow meant that the customer would receive a working implementation that could have immediate benefits aside from serving as software to be evaluated. For example,
implementing the MIMI functionalities for the administration workflow provided the customer with a usable software iteration that addressed user management, research project management, and scheduling. The customer could then not only provide feedback, but also use the software iteration while development shifted to the remaining workflows defined by the high-level metaphor.

<table>
<thead>
<tr>
<th>Functionality</th>
<th>People</th>
<th>Projects</th>
<th>Materials</th>
<th>Sessions</th>
</tr>
</thead>
<tbody>
<tr>
<td>create or schedule</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>view</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>edit</td>
<td>A</td>
<td>A</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>upload and download data</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>S</td>
</tr>
<tr>
<td>manage billing and accounting</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>B</td>
</tr>
<tr>
<td>record session details</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>B</td>
</tr>
<tr>
<td>generate performance reports</td>
<td>n/a</td>
<td>n/a</td>
<td>n/a</td>
<td>B</td>
</tr>
</tbody>
</table>

Table 5.2: The metaphorical workflow associations for MIMI’s functionalities by object. Values A, B, and S represent administration, business tasks, and scientific data management, respectively.

We were accustomed to having a clear sense of direction from our prior development experiences, and the workflows defined by the high-level metaphor helped steer our developmental experience on a well-defined developmental path. We did not know how we would exactly arrive at the end of the developmental path, but at least the workflows from the high-level metaphor gave us places to go. We received a more simplistic view of the MIMI system that was positioned above MIMI functionalities. This simplistic view gave us the ability to regard the MIMI system as three separate entities that would co-exist after development.

We also found that the high-level metaphor helped keep the customer on the same page with the developer. The customer could always know what was being developed
or tested without knowing low-level intricacies such as MIMI’s code organization. We found that this proved to be a valuable asset in the feedback stages of development, since the customer conveyed MIMI’s design flaws in a manner that allowed us to quickly pinpoint what was at fault. For example, during one feedback session on MIMI’s scheduling interface, the customer mentioned that: “The administrative process of scheduling a session for an imaging system was outputting error messages. Here is a list of the offending error messages.” With the high-level metaphor in place, the customer had the knowledge to provide detailed feedback in the context of a particular workflow of the metaphor. This in turn decreased development time during the subsequent revision process.

Low-level metaphor. The low-level metaphor helped us to break down the implementation requirements for each MIMI functionality on a file level. The low-level metaphor is a structural frame that goes through an assembly-line and transforms into a ready-to-use entity. We treated the structural frame as a file that contains Javascript and Hyper-Text Markup Language (HTML) code, i.e., what the user sees when using MIMI. As the structural frame progresses along the assembly line, it receives internal parts that, as a whole, represent content from MIMI’s existing information repository. The final step involves adding external modifications to the structural frame, and this translated well into the styling information contained within Cascading Style Sheets, or CSS. The low-level metaphor served as a guiding blueprint for organizing the implementation details for each MIMI functionality via separate files. Without the low-level metaphor, we would be inclined to place implementation details inside
one file, a decision that would slow-down development in the long-run.

5.2.4 Adhering to simplicity

We based MIMI’s development on “keeping things simple,” which directly translated to the simple design practice of XP. We found that implementing design specifications without contemplating future additions was a difficult, but also rewarding process. It is the standard practice of a good developer to plan for the future when implementing design specifications. However, we discovered that going through the adjustment period and harnessing a simple design path for MIMI’s development provided the following benefits.

*Synchronizing implementation to specification.* We found that undertaking a simple design path synchronized the implementation of MIMI software iterations with their corresponding functionalities. When implementing each software iteration, we were assured that the implementation details did not exceed the outlined specifications for each functionality. This served as a benefit for both us (the developers) and the customer. The luxury of having code match MIMI’s design specifications streamlined development since we did not have to spend time searching through code and deciphering the meaning of extraneous statements when implementing MIMI’s functionalities. Using the simple design philosophy also led to a “what you see is what you get” mind-frame for the customer. The customer knew that each MIMI software iteration corresponded to particular design specifications and did not have to evaluate unexpected features or question the developers about these features, which led to
minimal communication overhead.

*Unified file-naming convention.* With the goal of simplicity in mind, we utilized a unified file-naming convention for system files during MIMI’s development. –html and –css files comprised the logic for constructing the user interface. As a byproduct, we were able to identify a file’s type inside MIMI’s developmental environment without viewing its contents, which was useful for development.

*Coherent code organization.* We also discovered that adhering to the simple design philosophy led to coherent code organization within MIMI’s system files, which contained logic code. Each system file of MIMI has code that is divided across multiple functions: The Load Function is executed when a system file is initially accessed, and the Submit Function handles all access requests to a system file except for the initial access request. We felt that maintaining coherent code organization for system files facilitated both code reuse and modification during development.

### 5.2.5 Maintaining a customer relationship

Having access to the customer on-site, which is tantamount to the on-site customer practice of XP, proved to be an indispensable part of the developmental process. We were in a unique position of being stationed at the customer’s site throughout the bulk of the developmental process. We found that being directly on-site with the customer had the following benefits.

*Consistent communication.* We were able to communicate with the customer on a consistent basis during MIMI’s development, which proved to be advantageous.
We benefited from not dealing with fragmented means of communication. When we had questions or concerns about a specific design specification of MIMI, we met the customer in person, usually within a matter of minutes. We also did not have to make an ad hoc appointment and risk delaying a software iteration while waiting for the appointment date to arrive. Being close to the customer was also instrumental in acting as an implicit motivational factor for issuing software iterations on schedule. The luxury of quickly communicating with the customer was a two-way street. Not only could we meet with the customer when required, the customer could also choose to visit us. Knowing that the customer could see our work on an impromptu basis served as initiative for making frequent coding progress.

Timely feedback. When we worked on a software iteration of MIMI, timely feedback from the customer became the status quo. We found that timely feedback kept development from veering off-course. Timely feedback represented a counter-weight that suppressed our tendencies to implement functionalities in ways that did not co-incide with the customer’s intended interpretation. We learned to utilize constant customer feedback as a viable complement to releasing software iterations. The most useful feedback occurred when we met the customer after releasing a software iteration. We were able to offer our design perspectives and integrate them with feedback from the actual user’s point of view to make well-educated revisions to each software iteration. We discovered that having the customer “test-drive” a working feature in our presence led to changes that did not manifest in the original design specifications from the initial stage. We attributed this phenomenon as an example of a simple
proverb: “Do not judge a book solely by its cover.” More specifically, we learned that comprehensive specifications for a MIMI functionality are not tantamount to an ideal implementation when they do not include critical feedback from the customer’s hands-on assessment. We also noted that the quantity of critical feedback positively correlated with the interactivity scope of each software iteration. Working releases that incorporated features using static forms (e.g., making project requests) received less critical feedback than features that represented interactive applications (e.g., the session scheduling and scientific data downloading interfaces). Overall, we determined that obtaining timely feedback from the customer was the most useful aspect of MIMI’s developmental process. This is the case because the developers are implicitly creating the MIMI system vicariously through the customer when they are exposed to frequent and timely feedback.

5.3 Discussion

This chapter introduced a case study that described how the developmental methodology for an integrated informatics infrastructure known as MIMI translated into five specific practices of XP. Based on the positive developmental experiences from the case study, we conclude that these five practices of XP, as a whole, are a natural fit to act as the software developmental methodology of choice for developing such an integrated informatics infrastructure. With XP emphasizing small software releases over time instead of a holistic approach to software releases, development is tailored to adapting to the dynamic informatics needs of core facilities. We also conclude from
the positive results of the case study that each of the five relevant practices of XP is useful as an independent software developmental practice. For example, software developers may have frequent contact with the customer to facilitate even waterfall approaches to software development. Although MIMI’s developmental methodology led to a fully-functional system and is also natural fit to the five relevant practices of XP, the remaining XP practices must be evaluated in an appropriate setting before an informed decision is reached on the comprehensive viability of XP, i.e., whether using all twelve practices of XP is a worthwhile developmental methodology for creating an integrated informatics infrastructure for core facilities.
Chapter 6

Cost-benefit Analysis

6.1 Introduction

Cost-benefit analysis is a technique that evaluates both the positive (benefits) and the negative (costs) aspects associated with a process. The positive aspects, or benefits, associated with a process are of two kinds: tangible and intangible. Tangible benefits are positive aspects that are quantified with respect to some standard unit of measure. A typical cost-benefit analysis classifies tangible benefits using a monetary measuring standard such as the United States (US) dollar. Tangible benefits provide a point of reference that is consistent across different locales, e.g., tangible benefits which are expressed in US dollars are easily converted to a different monetary standard such as the Euro. In contrast, intangible benefits are positive aspects that are not reliably quantified with respect to some standard unit of measure. Also, the negative aspects, or costs, fall under two categories: recurring and non-recurring. Recurring costs represent the negative aspects that repeat over a period of time such as an annual
basis, while non-recurring costs are the negative aspects that require a one-time, up-front investment.

Dupuit’s analysis [24], which studied toll rates for bridge crossings, is regarded as the first example that clearly illustrated the purpose behind performing a cost-benefit analysis: to evaluate the attractiveness of a process as a solution for a particular task. “Attractiveness” is generally assessed by the overall benefits in comparison to an existing set of alternatives. The cost-benefit analysis exists to facilitate the decision-making concerning a process’s viability versus alternative options. Placing a monetary estimate on the attractiveness of a process provides a concrete value that serves as a consistent basis of comparison across processes. However, if the overall monetary benefit is insufficient or lacking, then the decision-making for adopting a particular process can transition to weighing the intangible benefits, although tangible monetary values are preferred for obvious reasons.

Although performing a cost-benefit analysis is well-defined, it is also difficult. Formulating plausible values for both the tangible benefits and costs of a process is non-trivial. The first challenge lies with determining what aspects to consider as sources of tangible benefits and costs, which often requires judicial use of the abstraction principle. Abstraction considers only those aspects of a process which are most relevant to the current context. Using abstraction when performing a cost-benefit analysis also eliminates sources whose monetary contribution may be negligible versus the amount of effort required to analyze them. Another challenge involves deciding on how to compute the monetary values of benefits and costs after their sources are identified. For example, when a person’s salary figures into a process’s
total cost, a decision must be made whether to include the profits that would accrue if the person was not involved with the process.

This chapter outlines the results of a cost-benefit analysis of using an integrated informatics infrastructure known as the Multi-modality Multi-resource Informatics Infrastructure (MIMI) for biomedical core facilities. MIMI is deployed at the Case Center for Imaging Research (CCIR), which is one of the leading core facilities in imaging. In Section 6.2, we describe the method used for the cost-benefit analysis and how this method was applied at the CCIR. In Section 6.3, we introduce a rough formula that estimates the net tangible benefits of using MIMI. In Section 6.4, we offer concluding remarks.

6.2 Cost-benefit Analysis of MIMI

Our cost-benefit analysis focused on directly accountable tasks from the viewpoint of the CCIR, which results in an under-estimate since all the users of MIMI receive a fraction of similar benefits on a regular basis.

In carrying out the cost-benefit analysis, we followed some existing examples in the literature. These include Grady’s analysis of an integrated telemental healthcare service for the military [34], Wang et al.’s cost-benefit analysis of electronic patient medical records [99], and Erdogmus’ approaches of cost-benefit analysis of software development [25]. We adhered to the general cost-analysis method, which we describe next.

Applying the general cost-analysis method to assess the attractiveness of a process
in comparison to the status quo involves four steps. Step one is calculating the tangible benefits of the process, i.e., the savings incurred from replacing the status quo with the process. Steps two and three are determining the process’s intangible benefits and calculating its recurring and non-recurring costs, respectively. Step four involves calculating the process’s net tangible benefits, which we accomplish by subtracting the process’s recurring and non-recurring costs from its tangible benefits.

It may seem obvious that using a highly integrated system such as MIMI would bring substantial benefits which can be translated to savings in expenditure for CCIR. However, a more precise analysis of the savings using the general cost-analysis method involved substantial work. Before carrying out the first step of the general cost-analysis method, we needed to find a common ground of comparison for MIMI (the process) and the status quo. Using either MIMI or the status quo addresses a core facility’s administrative and scientific workflows. The administrative workflow includes four tasks: scheduling imaging sessions, generating invoices, processing invoices, and compiling performance statistics to monitor resource usage. The scientific workflow consists of two tasks: acquiring scientific data and disseminating them to the relevant researchers. We met with staff members at the CCIR to determine an accurate and realistic estimation of the time spent on each task of the scientific and administrative workflows when using either MIMI or the status quo. We caution the reader that even though we tried to get as precise an estimation as possible, there are inherent reasons for some of the estimates to be based on rules-of-thumb only. The staff members at the CCIR also provided salary figures for the personnel who carry out the tasks of the administrative and scientific workflows. We used the salary figures together with
the time spent on each task of the workflows to estimate the tangible benefits from replacing the status quo with MIMI. We compared the annual salary expenditure for using MIMI versus the status quo for each task of the administrative and scientific workflows. We also combined our knowledge of MIMI and the status quo with information gathered from staff members at the CCIR to determine the intangible benefits of using MIMI. In addition, we used the abstraction principle to determine the recurring and non-recurring costs of using MIMI. For example, we included the annual cost of training users but excluded the annual cost of electricity usage.

We provide details on applying the general cost-analysis method at the CCIR in the sections below. In Section 6.2.1, we explain how the tangible benefits were calculated for each task of the administrative and scientific workflows. In Section 6.2.2, we elaborate on finding intangible benefits, which include an increase in administrative efficiency and a reduction in data management overhead. In Section 6.2.3, we discuss determining the recurring and non-recurring costs.

6.2.1 Tangible benefits

Using MIMI provides the CCIR with substantial savings versus the status quo. This savings is the culmination of replacing the status quo with MIMI to address the tasks of the administrative and scientific workflows. In Sections 6.2.1.1 and 6.2.1.2, we estimate the time spent when using either MIMI or the status quo for each of the tasks of the administrative and scientific workflows. We also include four diagrams [Figures 6.1-6.4] that show the necessary steps and the corresponding time estimates.
per working day for using either MIMI or the status quo to subsequently carry out all of the tasks of the administrative and scientific workflows. These diagrams follow the activity diagram specification of the Unified Modeling Language, or UML. A solid dot represents the initial state. Round-corner boxes denote activities. Round-corner boxes that do not contain time estimates in parentheses represent activities using either negligible or no administrative time. Solid arrows specify transitions between activities. Diamonds are decision nodes, which have outgoing solid arrows that are traversed only if the specified conditions in brackets are met.

6.2.1.1 Administrative workflow tasks

Scheduling imaging sessions. The status quo for scheduling imaging sessions involves three steps:

(1) A researcher contacts a CCIR staff member using phone or email communication.

(2) The researcher and staff member work out an amenable time.

(3) The staff member schedules an imaging session for the researcher and sends out a notification.

We will further analyze each step to estimate the administrative time spent for scheduling an imaging session. The first step is the responsibility of the researcher and does not occupy administrative time. The CCIR staff member takes a passive role until the researcher initiates communication. During the second step, the CCIR
staff member communicates with the researcher and also searches a calendar system for available time slots (the CCIR used Microsoft Outlook Calendar for scheduling imaging sessions). We estimate that the second step takes about 2 minutes of a CCIR staff member’s administrative time. This time measurement is obtained as the average of the estimates for phone and email communication in addition to the time estimate for searching the calendar system, which includes logging-into a computer and launching the calendar. The third step involves the entry of pertinent data such as a researcher’s name into the calendar system by the CCIR staff member. We estimate that the third step uses an additional 0.5 minutes on average since a valid time slot is already determined in step two. In total, we estimate that scheduling an imaging session takes approximately 2.5 minutes of a CCIR staff member’s time (2 minutes + 0.5 minutes). Based on the usage statistics from Section 3.3.1, we find that the CCIR averages about 4 imaging sessions per working day, which translates into an estimated 10 minutes of administrative time. Assuming the lower end of 260 working days per year and using $18 per hour for the salary of a low-level administrative staff member, the CCIR’s annual expenditure for scheduling imaging sessions with a low-level administrative staff member would be $780 (1/6 hour per working day x $18 per hour x 260 working days per year). However, a low-level administrative staff member cannot handle all the responsibilities of scheduling. A high-level scientific staff member with in-depth knowledge of imaging systems is involved in final decision making for tasks such as overseeing scheduling management and resolving scheduling conflict. Based on the prior practice at the CCIR, the high-level scientific staff member spends half of the working day on these tasks. Since the high-level scientific
staff member’s salary is around $90,000, the CCIR’s adjusted annual expenditure for scheduling imaging sessions is approximately $45,780 ($45,000 + $780).

Figure 6.1: UML diagram of the necessary steps and the corresponding time estimates per working day for using the status quo to subsequently carry out all of the tasks of the administrative workflow

Scheduling imaging sessions with MIMI displaces the administrative burden faced by staff members onto the researcher and leads to a savings in comparison to the status quo. With MIMI, the researcher logs-in and accesses a web-based calendar system to schedule an imaging session and does not burden a CCIR staff member. As a result, we find that using MIMI to schedule imaging sessions uses no administrative time, which translates into a negligible expenditure ($0) for the CCIR and tangible benefits of $45,780 annually ($45,780 - $0).

Generating invoices. The steps for generating an invoice with the status quo are as follows:

(1) The people who carry out an imaging session, whether they are CCIR techni-
cians or researchers, record session details inside a laboratory notebook. Session details can include a session’s time duration or the type of species that is scanned.

(2) Session details are transferred to a spreadsheet program and supplemented with information for determining the total cost of the corresponding imaging session.

(3) The total cost of the imaging session is calculated.

We will further analyze each step to estimate the administrative time spent for generating invoices. We approximate that the average time spent on step one is 1.2 minutes, which involves locating a laboratory notebook, accessing the correct page, and recording the session details. Based on the usage statistics from Section 3.3.1, we find that the CCIR averages about 4 imaging sessions per working day, which translates into an estimated 4.8 minutes of time. However, we assume that, on average, 25% of the people who perform step one classify as CCIR staff members, while 75% are researchers or research assistants. In turn, we calculate that the administrative time spent on step one is about 1.2 minutes per working day (4.8 minutes x 0.25). Step two, which includes logging-into a computer, launching a spreadsheet application, and entering session details and cost information, is the responsibility of a CCIR staff member. We estimate that step two utilizes about 2 minutes of administrative time per imaging session, which translates into 8 minutes of administrative time per working day. Step three involves logging-into a computer, launching a spreadsheet application, and entering formulas for cost calculation. We estimate that step three occupies about 2.50 minutes of administrative time on average, which equals 10 min-
utes per working day. We thus determine that the total administrative time spent per working day on generating invoices with the status quo is approximately 19.2 minutes (1.2 minutes + 8 minutes + 10 minutes).

![UML diagram](image)

Figure 6.2: UML diagram of the necessary steps and the corresponding time estimates per working day for using MIMI to subsequently carry out all of the tasks of the administrative workflow

Generating invoices with MIMI reduces the manual workload for staff members in comparison to the status quo, which in turn lowers the corresponding annual cost. Even when using MIMI to generate invoices, we find that step one’s 1.2 minutes are still a necessity due to auditing regulations, since the CCIR is required to maintain paper records for imaging sessions. A CCIR staff member then logs into MIMI and accesses and completes a session invoice, tasks that occupy about 8 minutes of administrative time per working day (2 minutes per session x 4 sessions per working day). Once the session invoice is completed, MIMI automatically calculates the imaging session’s cost, which occupies no administrative time. As a result, we find that
generating invoices with MIMI requires about 9.2 minutes (1.2 minutes + 8 minutes), or a savings of 10 minutes per working day versus the status quo (19.2 minutes - 9.2 minutes). Assuming the lower end of 260 working days per year and using $18 per hour for the salary of a low-level administrative staff member, using MIMI to generate invoices results in tangible benefits of $780 annually (1/6 hours per working day x $18 per hour x 260 working days per year).

*Processing invoices.* Processing an invoice with the status quo involves two steps:

1. A CCIR staff member references the total cost of an imaging session and the account number to be billed.

2. The staff member bills the relevant account number.

We will further analyze each step to estimate the administrative time spent for processing invoices. Step one includes logging-into a computer and accessing the correct spreadsheet records. We estimate that the total administrative time spent for steps one and two is 1.2 minutes, with each step occupying approximately 0.6 minutes. Based on the usage statistics from Section 3.3.1, we find that the CCIR averages about 4 imaging sessions per working day, which translates into an estimated 4.8 minutes of time.

Processing invoices with MIMI streamlines billing for staff members, which translates into a savings in comparison to the status quo. Using MIMI to perform step one requires about 0.4 minutes of administrative time to log-in and access a master invoice that contains the total costs and corresponding account numbers of imaging
sessions. However, step two still needs to be performed per imaging session, which uses 2.4 minutes of administrative time per working day. We thus find that MIMI provides a time savings of 2.0 minutes per working day in comparison to the status quo (4.8 minutes - 2.4 minutes - 0.4 minutes). Assuming the lower end of 260 working days per year and using $18 per hour for the salary of a low-level administrative staff member, using MIMI to process invoices results in tangible benefits of $156 annually (1/30 hours per working day x $18 per hour x 260 working days per year).

Compiling performance statistics. The CCIR compiles performance statistics to fulfill the requirements of grant applications and annual auditing procedures. We find that these performance statistics include queries such as the number of small animals processed or the prevalence of cancer-related research projects. Compiling performance statistics with the status quo involves two steps:

(1) Locating relevant documents in paper or electronic format.

(2) Going through the documents, extracting the pertinent information, and summarizing the performance statistics.

Based on the practice that compilation is performed on a monthly basis, we find that this amounts to a full-time job for two administrative staff members. Assuming that that an administrative staff member’s salary is approximately $36,000, the CCIR’s annual expenditure for compiling performance statistics with the status quo amounts to approximately $72,000 ($36,000 per staff member x 2 staff members).

Using MIMI, performance statistics are compiled automatically. The administrative time needed amounts to logging into MIMI, issuing a performance summary
query, and saving the results. Assuming that such queries are performed no more than several times a week, this incurs negligible time and expenditure for a staff member. As a result, using MIMI to compile performance statistics leads to tangible benefits of about $72,000 annually ($72,000 - $0).

6.2.1.2 Scientific workflow tasks

*Acquiring scientific data.* The CCIR contains imaging systems that create scientific data. Each imaging system is connected to its corresponding workstation computer via a dedicated link. The status quo for acquiring scientific data can be divided into two steps:

1. A CCIR staff member or researcher fills out and submits an imaging session’s metadata on a workstation computer. The metadata includes details such as the name of a researcher and the nametags of the species to be scanned.

2. Scientific data, at inception, travel from an imaging system and arrive at a workstation computer for storage.

We will further analyze each step to estimate the administrative time spent for acquiring scientific data. We estimate that the average time for step one is approximately 2 minutes per imaging session, since a CCIR staff member or researcher must log into a workstation computer and access/fill out/submit a metadata page. Based on the usage statistics from Section 3.3.1, we find that the CCIR averages about 4 imaging sessions per working day, which translates into an estimated 8 minutes of time. We also assume that step two occurs without setbacks such as imaging system
malfunctions and hence does not occupy administrative time. Assuming the lower end of 260 working days per year and using $18 per hour for the salary of a low-level administrative staff member, the CCIR’s annual expenditure for acquiring scientific data using the status quo with a low-level administrative staff member would be $624 (2/15 hour per working day x $18 per hour x 260 working days per year).

![UML diagram](image)

**Figure 6.3**: UML diagram of the necessary steps and the corresponding time estimates per working day for using the status quo to subsequently carry out all of the tasks of the scientific workflow.

With MIMI, a session’s relevant metadata is created automatically at the scheduling stage. The metadata does not have to be entered on a workstation computer, which amounts to negligible administrative time and tangible benefits of $624 annually ($624 - $0).

*Distributing scientific data.* Distributing scientific data into the hands of researchers and collaborators with the status quo can be accomplished in two ways: using portable media or file-sharing. We approximate that using portable media such as Digital Versatile Discs (DVDs) requires about 1.5 minutes of administrative time on average. Portable media must be loaded, the media-writing application must be started, and
scientific data must be selected by browsing folders on the workstation computer. Using file-sharing also occupies approximately 1.5 minutes of administrative time since the destination computer must be specified, access permissions must be set, and scientific data must be selected by browsing folders. Based on the usage statistics from Section 3.3.1, we find that the CCIR averages about 4 imaging sessions per working day, which translates into an estimated 6 minutes of time. Assuming the lower end of 260 working days per year and using $18 per hour for the salary of a low-level administrative staff member, the CCIR’s annual expenditure for distributing scientific data using the status quo with a low-level administrative staff member would be $468 (0.1 hour per working day x $18 per hour x 260 working days per year). We double this annual expenditure to $936 to account for re-issuing lost or misappropriated media.

Figure 6.4: UML diagram of the necessary steps and the corresponding time estimates per working day for using MIMI to subsequently carry out all of the tasks of the scientific workflow

In comparison to portable media or file-sharing, using MIMI for distributing scientific data also incurs about 1.5 minutes of administrative time. The Uploader application must be started, the Data Server Path value must be retrieved, and scientific data must be selected by browsing through folders (see Section 3.2.3.4 for more details). However, using MIMI eliminates re-issuing lost or misappropriated media, which translates into tangible benefits of $468 annually ($936 - $468).
6.2.2 Intangible benefits

We found that MIMI provides two intangible benefits: an increase in administrative efficiency and a reduction in data management overhead.

*Increase in administrative efficiency.* With MIMI, the CCIR’s administrative staff members have a central repository at their fingertips, which helps increase administrative efficiency. The central repository includes information such as the number of small animals scanned or the quantity of cancer related research projects. We found that with the status quo, such specific information was either not available due to ad hoc mishaps (e.g., paper records were lost before their data was entered into a spreadsheet application) or that calculating it became too time-consuming. MIMI’s central repository is used by the administrative staff to compile performance statistics for grant applications and audit requirements. Since all relevant information resides inside the central repository, we found that the CCIR staff members do not have to spend time on integrating information from multiple sources such as disjoint spreadsheets. The staff members then experience an increase in administrative efficiency since more tasks can be done in the same amount of time. For example, with MIMI, a CCIR staff member can compile multiple performance statistics for a grant application in one working day, which by CCIR personnel estimates, can facilitate the acquisition of an annual RO1 grant.

*Reduction in data management overhead.* Using MIMI entails accessing a web-based interface for all functionalities, which leads to a reduction in data management overhead. Researchers can access their scientific data in a manner that is uncon-
strained by space, time, or type of computer system. We found that accessing scientific data in such a manner mitigates overhead when needing to replace lost or misappropriated scientific data. Researchers can re-download scientific data through MIMI’s web-interface from the comfort of their home offices or labs instead of burdening core facility staff members. We also discovered that MIMI is an excellent tool for sharing information among the CCIR’s scientific community. MIMI provides researchers with the capability to share their scientific data with other researchers. MIMI’s sharing capability is more conducive to collaboration than sending scientific data through means such as an email attachment or regular mail. A reduction in data management overhead also applies to the administrative side. We noted that CCIR staff members can access MIMI’s central repository from any web-enabled computer, which allows them to compile performance statistics without location constraints, i.e., they do not have to be on-site at the core facility.

6.2.3 Costs

The costs for using MIMI are of two kinds: non-recurring and recurring. Non-recurring costs include the cost for development and implementation. They also cover initial hardware and software costs. Recurring costs include hardware upgrades and user training. MIMI’s development and implementation cost is approximately $100,000, with $50,000 for a full-time programmer, and $50,000 for a half-time supervisor for design and specification. MIMI also incurs a hardware cost of approximately $3,300. The hardware cost includes a primary server computer and installation fees.
Assuming that the server computer is replaced every 3 years results in an estimated annual cost of $1,100 for hardware upgrades. In addition, MIMI’s software cost is $0 since MIMI is built completely on open-source software that requires neither purchasing fees nor licensing costs. The estimation of the cost for user training is based on the assumption that the CCIR increases its user base by about 30 people annually. We include the 150 people who are currently using MIMI in the final training cost. We also assume that the annual salary of the training personnel is approximately $36,000, training personnel work 2,000 hours annually, and a training session lasts about two hours and trains 10 users. With these assumptions, we calculated the cost of a training session to be $36 per 10 users ($36,000 per year / 2,000 hours per year x 2 hours per training session), which equals about $3.60 per user. We then estimated the cost of training MIMI’s initial 150 users to be $540 ($3.60 per user x 150 users). We also determined that training 30 users per year incurs an annual cost of approximately $108 ($3.60 per user x 30 users).

6.3 Summary of Results

Table 6.1 shows a summary of the cost-benefit analysis of using MIMI at the CCIR. The annual tangible benefits and costs total $119,808 and $1,208, respectively. Using MIMI also incurs an initial total cost of $103,840. A rough formula that estimates the net tangible benefits of using MIMI for a period of \( n \) years, where \( n \geq 0 \), is:

\[
F(n) = 118600n - 103840
\]
<table>
<thead>
<tr>
<th></th>
<th>Benefits ($)</th>
<th>Costs ($)</th>
<th>Occurs annually</th>
</tr>
</thead>
<tbody>
<tr>
<td>Scheduling sessions</td>
<td>45,780</td>
<td>-</td>
<td>Y</td>
</tr>
<tr>
<td>Generating invoices</td>
<td>780</td>
<td>-</td>
<td>Y</td>
</tr>
<tr>
<td>Processing invoices</td>
<td>156</td>
<td>-</td>
<td>Y</td>
</tr>
<tr>
<td>Compiling statistics</td>
<td>72,000</td>
<td>-</td>
<td>Y</td>
</tr>
<tr>
<td>Acquiring data</td>
<td>624</td>
<td>-</td>
<td>Y</td>
</tr>
<tr>
<td>Distributing data</td>
<td>468</td>
<td>-</td>
<td>Y</td>
</tr>
<tr>
<td>Development</td>
<td>-</td>
<td>100,000</td>
<td>N</td>
</tr>
<tr>
<td>Initial hardware</td>
<td>-</td>
<td>3,300</td>
<td>N</td>
</tr>
<tr>
<td>Hardware upgrades</td>
<td>-</td>
<td>1,100</td>
<td>Y</td>
</tr>
<tr>
<td>Initial training</td>
<td>-</td>
<td>540</td>
<td>N</td>
</tr>
<tr>
<td>Further training</td>
<td>-</td>
<td>108</td>
<td>Y</td>
</tr>
</tbody>
</table>

Table 6.1: A summary of the cost-benefit analysis of using MIMI at the Case Center for Imaging Research

With three specific time points as input samples for the formula, we find that foregoing the status quo method and using MIMI over the time periods of one, two, and three years yields net tangible benefits of $14,760, $133,360, and $251,960, respectively. Figure 6.5 further expands the number of time points as input samples and shows the net tangible benefits of using MIMI over a continuous time period of up to twenty years following the initial launch. We see that the net tangible benefits increase linearly versus time progression.

### 6.4 Discussion

The results of the cost-benefit analysis of using MIMI for the CCIR’s informatics needs serve as evidence that a core facility gains net tangible benefits. About one million dollars can be saved within ten years. Again, this savings does not account for overhead savings provided by MIMI for the users in data transfer and sharing. Core
facilities can also use the results of the cost-benefit analysis as a helpful reference for estimating the tangible and intangible benefits and costs when adopting integrated informatics infrastructures which are similar to MIMI.
Chapter 7

Conclusions and Future Work

7.1 Conclusions

Although the existing MIMI system addresses the daily informatics needs of core facilities, it is also useful from a researcher’s point of view. The MIMI system serves as a catalyst for researcher-specific tasks such scientific data retrieval and information mining [27]. The MIMI system features the capability to seamlessly share scientific data with domain experts on a global scale via the internet and thus classifies as a direct facilitator of research. The existing MIMI system also inherently features information mining capabilities for research materials, which allows researchers to extract scientific data that apply to a particular research material over a period of time. Using the MIMI system’s information mining capabilities, researchers can perform comparative studies that lead to relevant discoveries. For example, they can retrieve the scientific data of a sample from two different datasets to locate essential information for detecting a gross tumor volume.
7.2 Future work

We now introduce future work that relates to MIMI. In Section 7.2.1, we describe modifications that can further MIMI’s deployment scope. In Section 7.2.2, we elaborate on how multiple MIMI systems can be linked into a single system and discuss the resulting impact. In Section 7.2.3, we discuss our plans to develop caMIMI, which is a MIMI system that will support the informatics needs of the multiple cores of the Case Comprehensive Cancer Center.

7.2.1 Modifications to improve deployment scope

Capturing general object attributes. Based on the deployment MIMI systems covered in Chapters 3 and 4, we find that deploying MIMI in multiple environments entails implementing specific features for each deployment environment. Creating a MIMI system that can act as an automatic drop-in solution for the needs of each deployment environment is not practical. Although a core facility adheres to its corresponding object model (e.g., the model defined in Section 2.3), accounting for all of the intricacies that are inherent to the core facility is difficult, since it is not possible to know the exact nature of the intricacies at hand. For example, in addition to using the standard object model attributes found in Chapter 2, the Case Center for Imaging Research tracks cancer center affiliations for ongoing research projects and membership areas of its registered user base, which we define as general object attributes.

To capture general object attributes inherent to each core facility, we will integrate installation wizards into MIMI. When installing the modified MIMI system, a core
facility will transition through screen prompts that capture the relevant information about object attributes and transform it into usable code. The primary benefit in this case involves minimizing development time since the bulk of the installation process falls on the core facility and not on the developer.

Managing analyzed scientific data. We will supplement MIMI’s existing feature-set for managing scientific data by introducing capabilities to manage analyzed scientific data [85]. Based on the information from Chapter 3, the Meta Server manages administrative functionalities, and the Data Server handles scientific data, so as to maintain a logical progression, we will introduce a third server: the Application Server. The Application Server will store analyzed scientific data and will only communicate with the Meta Server. The reasoning behind having the Application Server communicate only with the Meta Server is two-fold: increasing security and reducing redundancy. The Application Server will reside behind a hardware firewall and will receive protection from internet-native hazards. Also, since the Meta Server stores user profile objects that contain access privilege details, then tunneling all of the communication of the Application Server through the Meta Server will avoid the need to maintain redundant user profile objects on the Application Server. Of course, having the ability to store analyzed scientific data is useless without mechanisms for uploading and downloading analyzed scientific data. For uploading, we will use a funnel relationship that is tantamount to sending scientific data from a workstation computer to the Data Server. A researcher will upload analyzed scientific data from an internet PC to the Meta Server, and the Meta Server will send them to the Application Server.
For downloading, we will use a downloading interface that is identical to the one for scientific data. Again, both uploading and downloading will be optimized since they will use existing access privilege information on the Meta Server.

The Application Server will allow researchers to carry out research-specific tasks. Researchers will be able to distribute analyzed scientific data to collaborators without the hassles that are inherent to using email attachments and portable media such as DVDs. Secondly, since the Application Server will resemble the Data Server on the storage level. The Application Server will use a standard file-system for storage, and researchers will be able to perform cross-referencing of analyzed scientific data. For example, researchers who are working in-tandem on a research project will each retrieve analyzed scientific data for a single sample over multiple studies independent of cross-communication. As a result, the researchers’ productivity will increase because they will spend less time searching and more time concentrating on more significant tasks. Also, the Application Server’s uploading and downloading capabilities for analyzed scientific data will permit researchers to create important textual information that is easily retrieved by collaborators. For example, a researcher will be able to download analyzed scientific data, record relevant textual information inside a text file, and then upload the text file to the Application Server. In turn, the Application Server will feature annotation capabilities that will facilitate the interpretation of analyzed digital results with the help of other researchers.

Using external software. Since the MIMI system is web-based, we will introduce a capability for researchers to transparently access external software applications
through MIMI. It will be possible for a core facility to develop an external software application for viewing analyzed scientific data and integrate it directly into the MIMI system. Researchers will have the ability to utilize external software applications by using a standard web-browser that supports the Java plug-in, which is advantageous because it eliminates software version control and keeps researchers on the forefront of cutting-edge functionality. Since a web-based external software application must be downloaded from a server before it is used, updating the application’s master copy on the server will automatically lead to future downloads reflecting the latest software version with the most recent functionality.

7.2.2 Linking deployment systems

Although MIMI is competent as a stand-alone system, we will investigate broader information integration and retrieval tactics. We will transition the viewpoint from a specific MIMI implementation that applies to a single core facility to a more abstract viewpoint that considers multiple MIMI implementations as one cohesive system [29, 37, 55]. Since MIMI can reside at a plethora of core facilities world-wide, it is only natural to envision an environment that harnesses the information repositories from a large contingent of sources, e.g., treating core facilities that span three distinct domains such as proteomics, genomics, and imaging as one core facility. Having access to multiple MIMI information repositories will allow researchers to harmonize their analysis of specific types of research materials since different analyzed scientific data will be available at their fingertips. Linking multiple MIMI systems will also
impact information mining on a global scale. It will be possible to link MIMI systems that are deployed at core facilities that are exceptional over multiple domains to expose their relevant information repositories for searching within a globally accessible context. For example, a researcher in North America will be able to mine experimental scientific data across many specialized core facilities to aid the discovery of new information relationships. We next describe how to link multiple MIMI deployment systems together, which involves implementing one of two communication layers.

*Centralized communication layer.* We will consider deploying a new server that acts as the “brains” behind linking multiple MIMI deployment systems. The new server will communicate with each specific MIMI system using custom query wrappers, as is shown in Figure 7.1. For example, in the case that a researcher will query the new server for the analyzed scientific data of a single sample, the new server will communicate with each MIMI system (proteomics, genomics, and imaging), retrieve the correct analyzed scientific data from each MIMI system, and simultaneously introduce all of the analyzed scientific data to the researcher, which is done transparently.

![Figure 7.1](image)

*Figure 7.1:* How the centralized communication layer transparently combines multiple deployment MIMI systems
Decentralized communication layer. Another possible approach for implementing a communication layer involves cross-system communication. All MIMI systems will communicate with each other “behind-the-scenes.” For example, when a researcher will query the proteomics MIMI system to retrieve analyzed scientific data from multiple MIMI systems, the proteomics MIMI system will initiate communication with the remaining MIMI systems, retrieve their corresponding analyzed scientific data, and display them to the researcher. Once again, the researcher will be oblivious to the fact that the results comprise pieces from different information repositories.

Although MIMI is designed to tailor to the informatics needs of core facilities from domains such as imaging and proteomics, its scope can be expanded to cover other locations that face informatics challenges (for example, educational and financial institutions, and various types of businesses). Through the use of a combination of centralized and decentralized communication layers, deployment MIMI systems that span multiple disciplines will be linked to form a large-scale information network, or LIN [92]. The LIN’s MIMI systems will be accessible transparently for all information extraction needs, which will promote heterogeneous information mining.

7.2.3 caMIMI

The Case Comprehensive Cancer Center consists of seventeen different cores that span multiple disciplines [Table 7.1], and we plan on developing a MIMI system known as caMIMI that tailors to the informatics needs of many of these cores (caMIMI’s data model and corresponding object examples appear in Figure 7.2 and Table 7.2,
Table 7.1: The seventeen cores of the Case Comprehensive Cancer Center respectively). Drawing on our prior experiences with developing the Imaging and Proteomics MIMI systems, which are described in Chapters 3 and 4, we will develop MIMI systems that are core-specific. These systems manage scheduling, study billing, and other tasks that are associated with the core-level.

Figure 7.2: caMIMI’s data model, which follows the specifications of the Unified Modeling Language

In addition, we will implement a central MIMI system that will act as a hub for
managing information such as registered users and research project on the center-
level, which will eliminate redundancy on the core-level. Users from each core will log
into the core-specific MIMI system, which will communicate with the central MIMI
system to retrieve the users’ contact and project information for identification and
integration into core-specific tasks. The central MIMI system will also transparently
collect statistical information from the core-level MIMI systems and present it to
the cancer center’s staff members for an automated center-wide assessment. Unlike
with the Imaging and Proteomics MIMI systems, we will not only use an appropriate
developmental environment, but also an additional software application for version
control management.

<table>
<thead>
<tr>
<th>User</th>
<th>Project</th>
<th>Session</th>
</tr>
</thead>
<tbody>
<tr>
<td>login-name: abc123</td>
<td>project-id: 1221</td>
<td>session-id: 122113</td>
</tr>
<tr>
<td>user-id: 12</td>
<td>name: studying proteins</td>
<td>user-id: 12</td>
</tr>
<tr>
<td>name: jacek szymanski</td>
<td>grant-id: 3</td>
<td>project-id: 1221</td>
</tr>
<tr>
<td>phone: 10123456789</td>
<td>grant: 5R01HL12345101A2</td>
<td>service-id: 13</td>
</tr>
<tr>
<td>fax: 13334445555</td>
<td>program-id: n/a</td>
<td>status: completed</td>
</tr>
<tr>
<td>email: <a href="mailto:jacek@fake.edu">jacek@fake.edu</a></td>
<td>program: n/a</td>
<td>billing-policy: per hour</td>
</tr>
<tr>
<td>address: 123 Mayfield Rd</td>
<td>start: 01/01/08</td>
<td>charge: $2,100</td>
</tr>
<tr>
<td>city: Cleveland Heights</td>
<td>end: 12/31/10</td>
<td>start: 11:00am</td>
</tr>
<tr>
<td>state: Ohio</td>
<td>activity-type: n/a</td>
<td>end: 1:30pm</td>
</tr>
<tr>
<td>zip: 444106</td>
<td>user-ids: 10,12,13</td>
<td>actual-start: 11:05am</td>
</tr>
<tr>
<td>core-ids: 1,3,5</td>
<td>pi: jacek szymanski</td>
<td>actual-end: 1:13pm</td>
</tr>
<tr>
<td>status: pending</td>
<td>description: weekly studies</td>
<td>file-path: /data122113/</td>
</tr>
</tbody>
</table>

Table 7.2: Examples of objects from caMIMI’s data model and the values of their
attributes, which are expressed as attribute : value

Developmental environment. The developmental environment of choice for caMIMI
is Ruby on Rails (RoR) [31], which is an open-source web application framework.
RoR interfaces with external databases for storage management. It also unifies and
streamlines development through the use of the Model-View-Controller design pat-
tern. The “Controller” is an intermediary that separates back-end logic and data access (the “Model”) from presentation (the “View”).

Version control software. We will use the Ruby on Rails developmental environment in-tandem with version control software called Subversion [20]. Subversion will allow us to download the latest versions of caMIMI’s code and upload changes through a web-based interface, which promotes decentralized development.
Bibliography


