12-2012

Geographic Information Systems for Real-Time Environmental Sensing at Multiple Scales

Samuel Esswein
Clemson University, sesswei@clemson.edu

Follow this and additional works at: https://tigerprints.clemson.edu/all_dissertations

Part of the Geographic Information Sciences Commons

Recommended Citation
https://tigerprints.clemson.edu/all_dissertations/1074

This Dissertation is brought to you for free and open access by the Dissertations at TigerPrints. It has been accepted for inclusion in All Dissertations by an authorized administrator of TigerPrints. For more information, please contact kokeefe@clemson.edu.
GEOGRAPHIC INFORMATION SYSTEMS FOR REAL-TIME
ENVIRONMENTAL SENSING AT MULTIPLE SCALES

A Dissertation
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy
Forest Resources

by
Samuel Thomas Esswein
December 2012

Accepted by:
Dr. Christopher J. Post, Committee Chair
Dr. Jason O. Hallstrom
Dr. Gene W. Eidson
Dr. Elena A. Mikhailova
ABSTRACT

The purpose of this investigation was to design, implement, and apply a real-time geographic information system for data intensive water resource research and management. The research presented is part of an ongoing, interdisciplinary research program supporting the development of the Intelligent River® observation instrument. The objectives of this research were to 1) design and describe software architecture for a streaming environmental sensing information system, 2) implement and evaluate the proposed information system, and 3) apply the information system for monitoring, analysis, and visualization of an urban stormwater improvement project located in the City of Aiken, South Carolina, USA.

This research contributes to the fields of software architecture and urban ecohydrology. The first contribution is a formal architectural description of a streaming environmental sensing information system. This research demonstrates the operation of the information system and provides a reference point for future software implementations. Contributions to urban ecohydrology are in three areas. First, a characterization of soil properties for the study region of the City of Aiken, SC is provided. The analysis includes an evaluation of spatial structure for soil hydrologic properties. Findings indicate no detectable structure at the scales explored during the study. The second contribution to ecohydrology comes from a long-term, continuous monitoring program for bioinfiltration basin structures located in the study area. Results include an analysis of soil moisture dynamics based on data collected at multiple depths with high spatial and temporal resolution. A novel metric is introduced to evaluate the
long-term performance of bioinfiltration basin structures based on soil moisture observation data. Findings indicate a decrease in basin performance over time for the monitored sites. The third contribution to the field of ecohydrology is the development and application of a spatially and temporally explicit rainfall infiltration and excess model. The model enables the simulation and visualization of bioinfiltration basin hydrologic response at within-catchment scales. The model is validated against observed soil moisture data. Results include visualizations and stormwater volume calculations based on measured versus predicted bioinfiltration basin performance over time.
ACKNOWLEDGMENTS

First and foremost, I would like to thank my advisor Dr. Christopher Post for his guidance, patience, and encouragement during my graduate studies. I would also like to thank my committee members, Dr. Jason Hallstrom, Dr. Elena Mikhailova, and Dr. Gene Eidson for providing invaluable insight, perspective, and feedback during the development of my research. This research would not have been possible without the contributions of the Intelligent River® research team. In particular, I would like to express my gratitude to Dr. David White, Dr. Sebastien Goasguen, Dr. Daniel Hitchcock, and Christopher Bellamy, for providing software development expertise and field technical support.

I would like to acknowledge the financial support provided by Clemson University (Public Service Activities; Center for Watershed Excellence; Clemson Computing and Information Technology; School of Agriculture, Forestry and Environmental Sciences); the City of Aiken, South Carolina; and the National Science Foundation.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>TITLE PAGE</td>
<td>i</td>
</tr>
<tr>
<td>ABSTRACT</td>
<td>ii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>iv</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>vii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF ABBREVIATIONS</td>
<td>xi</td>
</tr>
</tbody>
</table>

## CHAPTER

### I. PREFACE

- Objectives                                                                 | 1   |
- Contributions                                                            | 2   |

### II. ARCHITECTURAL DESCRIPTION OF A STREAMING ENVIRONMENTAL SENSING INFORMATION SYSTEM

- Abstract                                                                 | 5   |
- Introduction                                                             | 6   |
- Related Work                                                            | 9   |
- Methodology                                                             | 10  |
- System Concerns                                                        | 13  |
- Stakeholders and Concerns                                              | 16  |
- Architectural Styles                                                   | 17  |
- Data Elements                                                           | 23  |
- Components                                                              | 24  |
- Architectural Viewpoints                                               | 27  |
- Conclusions                                                            | 47  |
- Acknowledgements                                                       | 48  |
- References Cited                                                       | 48  |

### III. TOWARDS ONTOLOGY-BASED DATA QUALITY VALIDATION IN LARGE-SCALE SENSOR NETWORKS

- 52
Table of Contents (Continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>52</td>
</tr>
<tr>
<td>Introduction</td>
<td>53</td>
</tr>
<tr>
<td>Background and Related Work</td>
<td>55</td>
</tr>
<tr>
<td>Existing Middleware Architecture</td>
<td>60</td>
</tr>
<tr>
<td>Semantic Sensor Network Middleware</td>
<td>64</td>
</tr>
<tr>
<td>Discussion</td>
<td>71</td>
</tr>
<tr>
<td>Conclusions and Future Work</td>
<td>75</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>76</td>
</tr>
<tr>
<td>References Cited</td>
<td>76</td>
</tr>
</tbody>
</table>

IV. BIOINFILTRATION BASIN MONITORING, MODELING AND VISUALIZATION USING GEOGRAPHIC INFORMATION SYSTEMS IN AIKEN, SC .............................................................. 80

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>80</td>
</tr>
<tr>
<td>Introduction</td>
<td>81</td>
</tr>
<tr>
<td>Materials and Methods</td>
<td>88</td>
</tr>
<tr>
<td>Results and Discussion</td>
<td>117</td>
</tr>
<tr>
<td>Conclusions</td>
<td>137</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>144</td>
</tr>
<tr>
<td>References Cited</td>
<td>144</td>
</tr>
</tbody>
</table>

APPENDICES .................................................................................. 149

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A: ArcGIS® Models for Bioinfiltration Analysis</td>
<td>150</td>
</tr>
<tr>
<td>B: Bioinfiltration Model Source Code</td>
<td>157</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Stakeholders and System Concerns</td>
<td>17</td>
</tr>
<tr>
<td>2.2 Sensor Ontology Module Metrics</td>
<td>36</td>
</tr>
<tr>
<td>2.3 Modified Sensor Ontology Modules</td>
<td>37</td>
</tr>
<tr>
<td>2.4 Modified Sensor Ontology Module Metrics</td>
<td>38</td>
</tr>
<tr>
<td>2.5 Dependency Structure Matrix</td>
<td>41</td>
</tr>
<tr>
<td>4.1 Selected soil physical and chemical properties for Fuquay sand (FuB) and Orangeburg loamy sand (OrA) for Figure 4-1 (Source: USDA/NRCS Soil Data Mart, 2012)</td>
<td>91</td>
</tr>
<tr>
<td>4.2 Selected geotechnical data for native soil samples</td>
<td>95</td>
</tr>
<tr>
<td>4.3 Selected physical properties for engineered media samples</td>
<td>97</td>
</tr>
<tr>
<td>4.4 Selected chemical properties for engineered media samples</td>
<td>98</td>
</tr>
<tr>
<td>4.5 Soil moisture monitoring assembly configuration by site and depth</td>
<td>102</td>
</tr>
<tr>
<td>4.6 Saturated hydraulic conductivity ($K_{sat}$) univariate analysis for native soil</td>
<td>118</td>
</tr>
<tr>
<td>4.7 Effective soil porosity ($VW_{C_{sat}}$) univariate analysis for all monitoring sites</td>
<td>122</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Component and Connector View</td>
<td>31</td>
</tr>
<tr>
<td>2.2</td>
<td>Tiered Architectural View</td>
<td>40</td>
</tr>
<tr>
<td>2.3</td>
<td>Module Interactions</td>
<td>46</td>
</tr>
<tr>
<td>3.1</td>
<td>High Level Architecture</td>
<td>57</td>
</tr>
<tr>
<td>3.2</td>
<td>Detailed System Diagram</td>
<td>68</td>
</tr>
<tr>
<td>3.3</td>
<td>Average Transit Delay Benchmark Results</td>
<td>72</td>
</tr>
<tr>
<td>3.4</td>
<td>Batch Size Benchmark Results</td>
<td>73</td>
</tr>
<tr>
<td>4.1</td>
<td>Map of study area, monitoring sites, and bioinfiltration basins</td>
<td>89</td>
</tr>
<tr>
<td>4.2</td>
<td>Map of soil sample sites and soil map unit boundaries</td>
<td>90</td>
</tr>
<tr>
<td>4.3</td>
<td>Bioinfiltration basin monitoring sites in Aiken, SC, USA</td>
<td>94</td>
</tr>
<tr>
<td></td>
<td>with (a) turf cover near intersection of Park, Union, and Fairfield streets</td>
<td></td>
</tr>
<tr>
<td></td>
<td>and (b) vegetated cover near intersection of Chesterfield, Richland, and</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Park streets.</td>
<td></td>
</tr>
<tr>
<td>4.4</td>
<td>Soil water content monitoring devices including the</td>
<td>103</td>
</tr>
<tr>
<td></td>
<td>a) Motestack data acquisition device and a b) soil moisture monitoring</td>
<td></td>
</tr>
<tr>
<td></td>
<td>assembly prior to infill of engineered bioinfiltration media (Image:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Christopher Bellamy, Clemson University, 2009)</td>
<td></td>
</tr>
<tr>
<td>4.5</td>
<td>Rainfall excess production and flow routing model diagram</td>
<td>115</td>
</tr>
<tr>
<td>4.6</td>
<td>Histogram of measured ($K_{sat}$) for native soil (Source: measured $K_{sat}$</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>from Schnabel Engineering, LLC, 2009).</td>
<td></td>
</tr>
<tr>
<td>4.7</td>
<td>Log normal QQ-Plot of measured saturated hydraulic conductivity ($K_{sat}$)</td>
<td>120</td>
</tr>
<tr>
<td></td>
<td>(Source: measured $K_{sat}$ from Schnabel Engineering, LLC, 2009).</td>
<td></td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td>4.8</td>
<td>Log transformed semivariogram for saturated hydraulic conductivity ($K_{sat}$) in native soils (Source: measured $K_{sat}$ from Schnabel Engineering, LLC, 2009)</td>
<td>121</td>
</tr>
<tr>
<td>4.9</td>
<td>Histogram for measured volumetric water content at all sites and depths between June 2011 and May 2012</td>
<td>123</td>
</tr>
<tr>
<td>4.10</td>
<td>Histogram for measured volumetric water content at site CRPS site 1, depth 12” between June 2011 and May 2012</td>
<td>123</td>
</tr>
<tr>
<td>4.11</td>
<td>Histogram for measured volumetric water content at site CRPS site 7, depth 12” between June 2011 and May 2012</td>
<td>124</td>
</tr>
<tr>
<td>4.12</td>
<td>Histogram for measured volumetric water content at site CRPS site 7, depth 18” between June 2011 and May 2012</td>
<td>124</td>
</tr>
<tr>
<td>4.13</td>
<td>Precipitation and soil water content for CRPS site 4 at multiple depths between August 2011 and May 2012</td>
<td>125</td>
</tr>
<tr>
<td>4.14</td>
<td>Precipitation and soil water content for CRPS site 4 at multiple depths between September 21, 2011 and September 24, 2011</td>
<td>126</td>
</tr>
<tr>
<td>4.15</td>
<td>Histogram of effective soil porosity ($VWC_{sat}$) without log transformation</td>
<td>128</td>
</tr>
<tr>
<td>4.16</td>
<td>Effective soil porosity ($VWC_{sat}$) versus time linear relationship by site and depth</td>
<td>128</td>
</tr>
<tr>
<td>4.17</td>
<td>95% Confidence intervals for effective soil porosity ($VWC_{sat}$) versus time linear relationship by site and depth</td>
<td>129</td>
</tr>
<tr>
<td>4.18</td>
<td>Linear model for effective soil porosity ($VWC_{sat}$) versus time</td>
<td>129</td>
</tr>
<tr>
<td>4.19</td>
<td>Residuals plot for effective soil porosity ($VWC_{sat}$) versus time</td>
<td>130</td>
</tr>
<tr>
<td>4.20</td>
<td>Normal probability plot for residuals of effective soil porosity ($VWC_{sat}$) versus time linear model</td>
<td>130</td>
</tr>
</tbody>
</table>
List of Figures (Continued)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.21</td>
<td>Simulation of Sept. 21, 2011 rainfall event for CRPS #4</td>
<td>134</td>
</tr>
<tr>
<td>4.22</td>
<td>CRPS. 50 year, 30 minute rainfall simulation</td>
<td>138</td>
</tr>
<tr>
<td>4.23</td>
<td>CRPS. 50 year, 30 minute rainfall simulation without effective porosity adjustment. Time: 30 minutes</td>
<td>139</td>
</tr>
<tr>
<td>4.24</td>
<td>CRPS site 1. 50 year, 30 minute rainfall simulation without effective porosity adjustment</td>
<td>140</td>
</tr>
<tr>
<td>4.25</td>
<td>CRPS. 50 year, 30 minute rainfall simulation with effective porosity adjustment. Time: 30 minutes</td>
<td>141</td>
</tr>
<tr>
<td>4.26</td>
<td>CRPS site 1. 50 year, 30 minute rainfall simulation with effective porosity adjustment</td>
<td>142</td>
</tr>
</tbody>
</table>
LIST OF ABBREVIATIONS

AOI: Area of Interest
BMP: Best Management Practice
BSM: Bioinfiltration Soil Media
CI: Cyberinfrastructure
GIS: Geographic Information Systems
GPR: Ground Penetrating Radar
HUC: Hydrologic Unit Code
LID: Low Impact Development
LIDAR: Light Detection and Ranging
OWL: Ontology Web Language
REST: Representational State Transfer
RDF: Resource Description Framework
SESIS: Streaming environmental sensing information system
WSN: Wireless Sensor Network
VWC: Volumetric Water Content
CHAPTER 1

PREFACE

The purpose of this investigation is to design, implement and apply a real-time geographic information system for data intensive water resource research and management. The research presented supports an ongoing, interdisciplinary research program supporting the development of the Intelligent River® observation instrument. The Intelligent River® allows end-users, researchers, educators, and policymakers to collect, share, and utilize a broad spectrum of hydrological and environmental data at ultra-dense temporal and spatial scales.

This dissertation is divided into three sections corresponding to the design, implementation, and application components of the information system. Chapter two provides an architectural description of the information system. Chapter three describes the implementation, and evaluation of the information system for the system concerns of performance and scalability. Chapter four applies the information system to support an urban stormwater improvement project employing Low Impact Development (LID) methods located in the City of Aiken, South Carolina, USA.

Objectives

Objective 1

Design software architecture for a streaming environmental sensing information system. The system should address the current and future needs of data-intensive water resource research and management.
Objective 2

Implement and test the software architecture.

Objective 3

Apply the information system for monitoring, analysis and visualization of a urban stormwater improvement project in the City of Aiken, South Carolina. Address the following concerns:

- Evaluate urban soil properties and their relation to hydrologic response.
- Monitor and evaluate long-term bioinfiltration basin performance using a sensor network based environmental monitoring implementation.
- Develop and apply a bioinfiltration basin predictive model to evaluate performance and support visualization of hydrologic responses.

Contributions

Chapter 1

This chapter provides a description of the objectives and contributions of the research.

Chapter 2

This chapter provides a formal architecture description of software for streaming environmental sensing information system. Includes:

- A definition of the system requirements and stakeholders.
• A description of the system from a functional viewpoint. This viewpoint includes discussion of defining system characteristics including the boundary, components, connecters, and environment.

• A description of the system from a development viewpoint. This viewpoint includes discussion of component dependencies, coupling among subsystems, and cohesion within subsystems.

• A description of the system from a data viewpoint. This viewpoint describes the data model and its decomposition into modules.

Chapter 3

This chapter describes the implementation and testing of a streaming environmental sensing information system. Includes:

• A description of the system implementation.

• An example of streaming processing for a quality-control use case.

• The results of a series of benchmarks to evaluate the performance and scalability of the implemented system.

Chapter 4

This chapter applies the information system for monitoring, analysis, and visualization of a green infrastructure urban hydrology improvement project for the City of Aiken, South Carolina. Includes:

• A characterization of the physical and chemical soil properties of the study site. The analysis includes an evaluation of the spatial structure of hydrologic properties of soils.
• The development and application of a novel metric for monitoring long-term trends in bioinfiltration performance.

• The development and implementation of a spatially explicit stormwater infiltration and excess runoff model to support bioinfiltration basin performance prediction and visualization.
CHAPTER TWO
ARCHITECTURAL DESCRIPTION OF A STREAMING ENVIRONMENTAL SENSING INFORMATION SYSTEM

S. T. Esswein1*, J.O. Hallstrom2, C. J. Post1, G. W. Eidson3, and E.A. Mikhailova1

1School of Agricultural, Forest, and Environmental Sciences, Clemson University
2School of Computing, Clemson University
3Department of Biological Sciences, Clemson University

Clemson, SC 29634, USA

*Corresponding author. Current address: Department of Forestry and Natural Resources, Clemson University, 214 Lehotsky Hall, Clemson, SC 29634, USA. Tel.: 540-272-6254, Fax: 864-656-3304.

Email address: sesswei@clemson.edu (S.T. Esswein).

Abstract

Data intensive science is heralded as the next paradigm in scientific discovery. Nowhere is this transformation more evident than with environmental sensing applications. The vast majority of environmental sensing information already spends its entire life in a digital format. The next technological leap involves transferring sensing information from its point of acquisition directly into software applications and tools for scientists and decisions makers. Achieving this capability involves a complex array of software, hardware, networks, and people. At the center of the cyberinfrastructure is a streaming, environmental sensing information system. This information system,
supporting a large number of entities and complex interrelationships, requires software architecture that manages complexity through a carefully orchestrated separation of concerns. This modularizes the problem into tractable components. Software design has long devoted attention to this modularization process. The extent of modularization is often described using the software structural qualities of cohesion and coupling. This paper provides an architectural description of a streaming, environmental sensing information system, with an emphasis on characterization of cohesion and coupling. A clearer picture of the problem and solution are achieved through an evaluation of high-level interactions between distributed components and shared data models. Results from this research support software design decisions and facilitate comparison among alternative architecture and data model approaches.

**Keywords:** coupling, cohesion, data flow, software architecture

**Introduction**

Digital information is growing faster than it can be managed, analyzed, and understood (Bell 2009). This growth is occurring across the digital spectrum: from social networking websites to environmental sensor networks. Bell (2009) and others (Gray et al. 2007; Szalay and Gray 2001) describe a pressing need for new technologies capable of leveraging this data for scientific discovery. This technology needs to address the four activities of data intensive scientific discovery: (1) capture, (2) curation, (3) analysis, and (4) visualization (Gray 2007; Hey et al. 2009). The past decade has seen major advancements in cyberinfrastructure capable of supporting these activities, but there are many remaining challenges and opportunities (Foster and Kesselman 2006; NSF
Cyberinfrastructure Council 2007). Environmental sensing is a key driver of data intensive scientific discovery, producing petabytes of data every day (Yang et al. 2010). Low power computing and networking technology advancements are increasingly linking environmental sensing instrumentation to the Internet, opening up opportunities for information sharing and real-time analysis. Despite their recent advancements, sensor technologies have had a difficult time finding footing among conventional Web technologies. This can be attributed to the markedly distinct models of information, interactions, and applications characteristic to sensing information systems.

Cyberinfrastructure (CI) for streaming environmental sensing supports the activities of data intensive science. An environmental sensing CI can be decomposed into four tiers: 1) sensing fabric, 2) backhaul and communications, 3) middleware, 4) application. The sensing fabric captures data and interfaces with the communication tier. The communication tier supports physical network connectivity between all participants who are potentially located over long distances or in remote areas. Middleware directs high-level software interaction, controlling application complexity and enabling high performance and reliable operation. The application tier describes the curation, analysis, and visualization activities of data intensive science. A streaming environmental sensing information system (SEVIS) provides the foundation for interoperability among components of the CI, which links sensor systems to streaming applications.

The SESIS implements the middleware tier and a subset of the application tier. Its purpose is to simplify the obstacles associated with enabling streaming sensor systems and to reduce the complexity associated with developing streaming applications. To
fulfill these requirements, the system must tolerate failures and adjust to changes in information processing demands. Dependencies among participants should be carefully controlled in order to support highly dynamic collaborations. Routine functions, like maintaining observation and provenance records, should be provided to ensure that no data goes missing. Additionally, the system should provide a framework for basic quality control checks with the capability to annotate streaming observation data. All processing steps should be traceable through auditing procedures.

The SESIS involves a large number of components with complex interrelationships. These fundamentally distributed components are separated not only by memory access, but also by geographic and network hop distances. The dynamics of interaction among the distributed system components is a key point of departure from conventional software architecture and is also a primary concern for system architects. The structural software qualities of component coupling and cohesion are important to this interaction. Coupling describes interdependency among components and is considered an undesirable quality of software, while cohesion describes the relatedness of elements within a component and is considered a desirable quality. These qualities, which guide software decomposition, enable system parallelization and reduce complexity through the principle of separation of concerns.

This paper provides an architectural description of a SESIS. The description documents the design elements as well as the architectural decisions and rationale used during the design process. The SESIS described is an operational prototype supporting the Intelligent River research program (Eidson et al. 2010). The paper is organized as
follows: Section II provides a survey of related work; Section III describes the methodology guiding the architectural description; Section IV includes system concerns; Section V identifies Stakeholders and Concerns; Section VI describes constituent architectural styles; Sections VII and VIII include a description of the architectural components and data elements; Section IX presents a series of architectural views that document the elements and relations of the architecture. Conclusions are drawn in Section VI.

Related Work

The past decade has seen the development and implementation of numerous CI initiatives intended to perform environmental sensing at large-scales. Examples include GEOSS for Earth Observation (Butterfield et al. 2008), GEON for the Geosciences (Zaslavsky et al. 2005), and NEON and LTER for Ecological Sciences (Lowman et al. 2009; Karasti et al. 2006). While the scope and goals of these efforts vary, common themes exist. These include the need for stronger data stewardship, expressive metadata, data integration tools, and high performance computing resources to analyze data (Yang et al. 2010). Borgman et al. (2007) describe the architectural requirements associated with environmental sensing based on the experiences of the Center for Embedded Network Sensing (CENS).

The Open Geospatial Consortium (OGC)\(^1\) provides a family of models, encodings, and services that support environmental sensor webs. The Sensor Web Enablement (SWE) initiative allows interoperable and scalable service-oriented networks

\(^1\) [http://www.opengeospatial.org](http://www.opengeospatial.org)
\(^2\) [http://www.rabbitmq.com](http://www.rabbitmq.com)
for heterogeneous sensor systems and client application (Reed et al. 2007). The SWE specifications relevant to this discussion include two languages—Observation & Measurements (O&M) and Sensor Model Language (SensorML)—and four services—Sensor Observation Service (SOS), Sensor Planning Service (SPS), Sensor Alert Service (SAS), and Web Notification Service (WNS). Planned additions to SWE include a Sensor Event Service and an Event Pattern Markup Language (Bröring et al. 2011). These additions support a standardized approach to event stream processing and provide functionality similar to the SESIS described in this paper. The SWE approach is based on a Service Oriented Architecture (SOA) style implemented with Web Service technologies.

Methodology

Wand and Weber (1990) provide a series of formalisms useful as a basis for architectural discussions on system decomposition. The following informal overview begins with the concepts of things, events, and couplings. A change of state in a thing constitutes an event. A thing has an ordered history of events. Two things are considered coupled if “at least one of the things’ history depends upon the other thing’s history” (Wand and Weber 1990, pg. 1284). A system is defined as the set of things that cannot be partitioned such that a thing is not coupled to at least one other thing. Components in a system can be partitioned into subsystems. The system’s environment describes things that are coupled to components in the system, but not considered part of the system. The delineation of a system’s environment and its decomposition into subsystems are central architectural design tasks. The goal of decomposition is to group functionally similar
components into *tightly coupled*, cohesive subsystems, while simultaneously minimizing coupling between subsystems (*loosely coupled*). This goal corresponds with the *principle of separation of concerns*, which allows attention to be isolated and focused on a single aspect of a larger problem (Dijkstra 1974).

Clements et al. (2011, pg. 1) define architecture as “the prudent partitioning of a whole into parts, with specific relations among the parts.” This echoes the sentiment of most widely used definitions, although the degree of specificity may vary (Garlan and Shaw 1994; Perry and Wolf 1992). Formal discussions describe architectures in terms of elements and relations. Garlan and Shaw (1994) identify two architectural elements: components and connectors. Perry and Wolf (1992) add data as a first class architectural element. This documentation borrows from both approaches and, following the recommendation of Clements et al. (2011, pg. 148), presents modules and components as separate elements. Components are the system’s principal processing units, as seen from a runtime perspective. Modules are implementation units and are analogous to a component type. A *module* provides a closer correspondence to the implementation of a system, while components provide a conceptualization closer to our intuition. Modules and components may represent the same concepts, but from different perspectives. Connector elements describe the interactions among components. A connector can abstract a complex relationship between components, including relationships involving an intermediate entity. Connector entities have the constraint that no transformations occur on the data that passes through them. This distinction between components and connectors accurately accounts for sources of dependency among elements.
Architectural documentation is presented using the methodology described in the ISO/IEC 42010 standard for the architectural description of systems and software engineering (ISO 2010). This standard specifies the manner in which architecture descriptions should be organized and expressed. The architectural description identifies system concerns and stakeholders; then reviews the architectural styles used by this system. Styles represent common arrangements of elements and relations. Styles are analogous to design patterns in OO programming. High-level architectural components of the system are identified and described based on the roles they play in various architectural styles. The architecture of the SESIS is exposed using the documentation conventions of viewpoint and views. Views are a representation of a set of system elements and the relationships associated with them (Clements et al. 2011, pg. 22). A view conveys architecture from a particular viewpoint. A viewpoint frames a set of system concerns and stakeholders, and presents a particular perspective on architecture. An ISO 42010 conformant architectural description includes viewpoints that cover all the identified system concerns. The documentation provided here does not provide sufficient content to be fully conformant as an ISO architectural description.

This architectural description is primarily intended for an audience of developers and architects of alternative environmental sensing information systems. It provides a basis for application development with the current implementation and guides future evolution of the system. A formal description invites clearer comparison with alternative approaches and the opportunity to share design decisions.
System Concerns

This section defines the system concerns and the rationale behind their inclusion into the architectural description. In later sections, these concerns are evaluated against an instantiation of this architectural design to evaluate design decisions. System concerns are influenced by previous experience with environmental sensing applications, stakeholder inputs, and functional requirements of analysis and visualization tools.

Reliability

Reliability describes the susceptibility of the system to failures. Failures may arise from hardware, network, or software, which correspond to components, connectors, and data elements in an architectural description. In this system, component failures are a common occurrence and must be anticipated. While non-architectural factors influence reliability, robustness to failure ultimately stems from architectural decisions. Eliminating single points of failure, supporting fail-over conditions, and incorporating system diagnostic capabilities improve reliability.

Performance

This analysis treats performance as a catchall term for system throughput, bandwidth, and latency. Throughput describes the rate per unit time of information that can pass through the system. The minimum possible throughput corresponds with the latency. Available bandwidth determines throughput with the caveat that throughput time cannot improve latency values. In this system, performance is associated with the transit delays exhibited by individual observation events as they move from sensor to data consumer (quantified as transit delay). Information processing is required to have
bounded-time transit delays, e.g., real-time guarantees. The transit delay from sensor to
data consumer is influenced by factors not generally considered part of an information
system, e.g., backhaul communications link. Performance evaluations may restrict
comparisons to lower latency and higher throughput TCP/IP connections found on the
Internet or local area networks. This approach isolates architectural performance
properties from implementation specifics. A detailed description and metrics for
performance are provided in Esswein et al. (2012). Performance concerns can influence
design decisions associated with other concerns, e.g., scalability and reliability.

**Scalability**

Scalability is the system’s ability to adapt to changing demands. Despite its
widespread use, the term scalability lacks consensus on a universal, rigorous definition
(Bondi 2000; Duboc et al. 2006; Hill 1990). To reduce ambiguity, we describe scalability
as maintaining system concerns of performance and reliability with the addition and
removal of system components. For example, as the number of sensor systems change
over the system’s lifetime, a corresponding change to other components may be
necessary to maintain performance and reliability. A scalable system is one that
accommodates these fluctuations without architectural changes or software re-
engineering. Scaling resources to demands is necessary for maintaining long-lived and
highly dynamic sensing applications. Efficient resource use is a particularly important
consideration if the system is deployed in a cloud-computing environment. In this case,
operational cost is linked directly to resource utilization. From an architectural design
perspective, component coupling strongly influences scalability. As the number of

components grows, so do the number of component interactions. Architecture determines the nature of this growth rate. If the interactions exhibit high coupling, then it becomes difficult or impossible to separate processing concerns, thus preventing the distribution of processing.

**Modularity**

Modularity refers to the degree of separation of concerns achieved during system modularization. Modularity is promoted by reducing coupling between components and increasing cohesion within components. It allows the construction of complex software systems and is fundamental to the concerns of extensibility, scalability, and ease of use.

**Extensibility**

Extensibility is the ability to adapt a system to meet future demands. An extensible system allows components to be added, replaced, or updated without impacting its architecture. It is closely tied to the concept of modularity. Extensibility can be enhanced through architectural decisions.

**Ease-of-Use**

Ease-of-use is a quality that describes the stakeholders’ difficulty in using the system. From an architectural standpoint, ease-of-use is closely linked to modularity. An effective separation of concerns is more likely to yield a decomposition that coincides with a user’s intuition. A modular system isolates complexity from users. Stakeholders need not understand the details of the entire system to effectively utilize it.
Security

A secure system protects information and system components from unauthorized access, use, or disruption. This ensures information confidentiality, integrity, and availability. Security is accomplished through authentication and access control mechanisms.

Stakeholders and Concerns

Stakeholders and associated concerns are described in Table 2-1. The six stakeholder categories group the target audiences. Observers are split into three categories based on data analysis needs and the user skill set. Low-level observers correspond with software developers or scientists with real-time data requirements. These users require direct programmatic integration with real-time data streams. Applications are anticipated as custom developed at this level. Mid-level observers still require direct access to real-time streams, but access this information through existing tools or a high level Application Programming Interface (API). This category of users includes scientists and data modelers. The high-level observers include scientists, decision-makers, and the general public. These users do not require direct access to data streams. High-level observers access data through a high level interface, e.g., a browser based web portal. Users in this group are expected to have minimal knowledge of the underlying architecture. The operator stakeholder is concerned with the overall function of the system. This group includes information system managers and scientists responsible for
Table 2-1. Stakeholders and System Concerns.

<table>
<thead>
<tr>
<th>Stakeholder</th>
<th>Description</th>
<th>Reliability</th>
<th>Performance</th>
<th>Scalability</th>
<th>Modularity</th>
<th>Extensibility</th>
<th>Reusability</th>
<th>Ease-of-Use</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operators</td>
<td>System Operators</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>Low-level observer</td>
<td>Advanced, real-time data users</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mid-level observer</td>
<td>Real-time data users</td>
<td>X</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>High-level observer</td>
<td>General data users</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Data owners</td>
<td>Own sensor components</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Field technicians</td>
<td>Maintain sensor components</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

system function and data integrity. Data owners are users with direct responsibility over monitoring implementations, e.g., scientists and decision-makers. Field technicians maintain monitoring implementation, and are expected to have minimal interaction with the software components of the system.

Architectural Styles

Publish/Subscribe

The Publish/Subscribe (PS) style specializes the Event-Based Integration (EBI) architectural style (Garlan and Shaw 1994). Participants include Publisher(s), Subscriber(s), and a broker (hereafter referred to as a Communications Mediator [CM]). Architecturally speaking, there is no limit to the number of possible publishers or subscribers. While a CM appears to publishers and subscribers as a single entity, it may be clustered to address performance or reliability concerns. Logically, the interaction
bears similarity to the Observer object-oriented (OO) design pattern described by Gamma et al. (1995). In the Observer pattern, one or more Observers are notified when an event occurs on a Subject. An Observer specifies an interest in receiving notifications by registering itself with the Subject. When an event occurs on the Subject, a notification message is sent to all registered Observers. The result is a one-to-many relationship between Subject events and Observer notifications. Gamma et al. (1995, pg. 299) propose adding a ChangeManager entity to enable many-to-many relationships between Subjects and Observers, i.e., multiple Subjects may notify many Observers. A ChangeManager assumes the Mediator role (Gamma et al. 1995, pg. 273) between the Publisher and Subscriber. The ChangeManager is responsibility for 1) registering and unregistering Observers and 2) routing notifications to Observers. The PS style extends the Mediator-Observer OO approach into a distributed programming paradigm. The Publisher role assumes the responsibilities of the Subject, while the Subscriber assumes the responsibilities of an Observer. The CM assumes the role of ChangeManager and may, depending on the implementation, provide sophisticated message routing mechanisms, e.g., topic-based or content-based.

Topic-based message routing requires the Publisher to associate a topic string with each notification message. The string includes a series of words delimited by a special character, e.g., a period. A Subscriber expresses interest in a subset of messages based on a string-matching pattern, which may include wildcard patterns. The arrangement of words is up to the Publisher. A hierarchical convention is commonly used, arranging words in a manner similar to that of the Domain Name System.
(Mockapetris 1987). The CM is responsible for routing messages to Subscribers based on topic pattern matches. Descriptions of other message routing approaches, along with a review of implementation examples, can be found in Eugster et al. (2003).

The primary benefit of PS is the promotion of *loosely coupled* interactions. Eugster et al. (2003) describe this coupling in terms of three dimensions: space, time, and synchronization. Space decoupling indicates that participating Publishers and Subscribers need not be aware of one another. The message routing facility is used as an abstraction over participant identities. Space decoupling is partially offset by dependencies introduced by the message routing convention. For systems with an unknown number of participants, the trade-off favors PS since coupling is not influenced by the addition (or removal) of Publisher or Subscriber components. Time decoupling removes the requirement that both parties be present simultaneously during an interaction. A CM may store events on behalf of Subscribers, allowing them to be received at a later time. Synchronization decoupling requires that the execution of one party not be blocked while waiting on a response from the other party. In PS interactions, notification messages are always *pushed*-to, rather then *pulled*-by, interested Subscribers. Garlan and Shaw (1994) describe this as *implicit invocation* of procedures in other components. This reduces interactions per message compared with a request/reply approach. More importantly, it puts the Publisher, by way of the CM, in control of the Subscriber’s application state. Thus, the Subscriber’s application state is determined by events occurring on the Publishers. This explains why PS is subsumed under the EBI architectural style. In addition to the dimensions identified above, PS reduces interface coupling by providing a
uniform interface (publish, [un]subscribe, and notify) with easily understood semantics. With some exceptions, e.g., semantic matching (Hasan et al. 2012), PS implementations are payload agnostic and do not influence data coupling.

**Worker Queue**

The Worker Queue (WQ) architectural style is introduced as a specialized EBI Style. A queue refers to a data structure that provides sequential access to data with a first-in-first-out constraint. Message queues may be used as a backing data structure for EBI implementations, e.g., RabbitMQ\(^2\). Functionally, a WQ is similar to a PS and shares the same decoupling properties. Eugster et al. (2003) describe a similar message queue interaction style, but include a synchronous interaction constraint that does not apply to the WQ described here. Hohpe and Woolf (2003) use the term messaging to describe a class of asynchronous enterprise integration styles that include PS and WQ styles. Their description includes a comprehensive catalog of messaging elements. This discussion deviates from their description by focusing on a specific organization of these elements, which produces an architectural style.

Participants of the WQ include Producer(s), Worker(s), and a CM. As with the Publisher in a PS style, a Producer encodes events into notification messages. However, unlike a PS, the event results in a single notification message. The result is a one-to-one relationship between Producer events and Worker notifications. Notification messages are delivered to one of many possible Workers on a first-come, first-serve basis according to its assigned queue. The CM routes incoming messages to a queue based on a simple

\(^2\) [http://www.rabbitmq.com](http://www.rabbitmq.com)
identifier convention. While the PS and WQ styles appear outwardly similar, the WQ style goals are quite different. A WQ is used to parallelize tasks to a pool of Workers. The size of the pool determines the bandwidth available to transmit messages. Workers can be scaled out (or in) to maintain consistent message throughput rates under variable traffic conditions while incurring negligible increases in processing overhead (Esswein et al. 2012). A WQ can address reliability by requiring Workers to explicitly acknowledge task completion with the CM. In this scenario, if a task assigned to a Worker is not completed within a specified time interval, it will be reassigned to a different Worker.

The interface semantics differ slightly from the PS style: *enqueue, attach, detach, and notify*. However, the interface coupling remains low. Additionally, as with the PS style, the Producers act as the engine of the Workers’ application state.

### Representational State Transfer

The REST architectural style was first introduced by Fielding in his dissertation (2000), and again in Fielding and Taylor (2002). Fielding was the principal author of the HyperText Transfer Protocol (HTTP) specification (Fielding et al. 1999). The REST style communicates the architecture of the Web, based on Fielding’s characterization of the optimal arrangement of architectural elements, relations, constraints, etc. A brief summary of the REST style as it relates to this architecture is provided in this section.

REST incorporates multiple architectural styles, but is fundamentally a Client/Server style (Garlan and Shaw 1994) with two primary participants: the Client and Server. REST may include intermediary participants, e.g., caches, gateways and proxies. The Server role maintains no client state and must provide a uniform interface to the
resources it provides. The uniform interface reduces interface coupling. Additional constraints describe the uniform identification of resources, manipulation of resources through representations, the use of self-descriptive messages, and the use of hypermedia as the engine of application state. The REST style is most appropriate for large-grain data transfers (Fielding 2000). The Client/Server approach is a pull-based interaction, which involves a minimum of two interactions per transaction. The primary advantage is that the client has control over what (and when) information is received. Unless caches are employed, all parties must be present during an interaction. Interaction is generally synchronous, but asynchrony is possible if the client employs callback mechanisms, e.g., Asynchronous JavaScript (AJAX).

The constraint that hypermedia should determine application state has implications for component coupling. The constraint applies when hypermedia enabled resource representations are used by the client. Hypermedia representations provide a listing of possible state transitions, accompanied by embedded instructions. The instructions provide the necessary information for the client to initiate the next state transition when a particular transition is selected, e.g., an HTTP get request. An operator, usually a human, is tasked with triggering a specific state transition. From an architectural perspective, the hypermedia is responsible for providing the client with a subset of identities. The client requires the identity of a single resource to begin using the application, e.g., a homepage configuration setting. Once the application is bootstrapped with the identity, the hypermedia guides the acquisition of new identities at runtime. This approach eliminates the need for a client to store large numbers of identities, thereby
reducing identity coupling. This property has played a significant role in allowing the Web to scale to include over 1 trillion unique resources (Alpert and Nissan 2008).

**Repository Style**

The Repository architectural style describes a class of styles involving datastore components that retain large collections of persistent data (Clements et al. 2011, pg. 178). Data accessor components can read and write to this datastore. The SESIS architecture uses the Repository style to store and provide access to metadata and observation data. Examples of Repository datastores include relational database management systems, document stores, or key-value stores. Participants include the datastore and its accessors, with accessors initiating the interaction. Implementation specifics determine interface, data, and interaction coupling.

**Data Elements**

The Resource Description Framework (RDF) is used as a basis for data interchange in the SESIS architecture. RDF is a suite of recommendations for data representation from the World Wide Web Consortium (W3C)\(^3\). RDF provides the foundation for the Semantic Web. In addition to RDF, this architecture relies on standards built on top of RDF. These include: 1) RDFS\(^4\) for describing groups of related resources and the relationships between the resources, 2) the OWL\(^5\) languages for expressing vocabularies and ontologies, and 3) the SPARQL\(^6\) language for querying RDF

\(^3\) [http://www.w3.org/RDF/](http://www.w3.org/RDF/)

\(^4\) [http://www.w3.org/TR/rdf-schema/](http://www.w3.org/TR/rdf-schema/)

\(^5\) [http://www.w3.org/TR/owl2-overview/](http://www.w3.org/TR/owl2-overview/)

\(^6\) [http://www.w3.org/TR/rdf-sparql-query/](http://www.w3.org/TR/rdf-sparql-query/)
datasets. Although RDF is intended for machine interpretation, it can be serialized into human-readable, text-based formats. RDF is based on the assertion of a statement in the form of a *triple*, which includes a subject, predicate, and object. Triples are comprised of resources, which can be Uniform Resource Identifiers (URI) or literal values (primitives). This syntax allows statements to link together with other statements to form graphs.

The Web Ontology Language (OWL) allows the expression of ontologies using RDF. Ontologies are an explicit conceptualization of a knowledge domain (Gruber 1995). Ontologies describe a vocabulary of terms with explicitly defined relations and are useful for sharing information among a community of users. Ontologies may be combined with other ontologies to build extensive vocabularies and encourage the sharing of concepts between knowledge domains. The Semantic Sensor Network (SSN) ontology is used by this system to express environmental sensing concepts. The SSN ontology was developed as part of an Apache incubator project completed in 2011 (Compton et al. 2012). The SSN ontology represents a collaborative effort to harmonize existing sensor ontologies with the Open Geospatial Consortiums suite of Sensor Web Enablement standards. Other ontologies used by this system include the DOLCE upper level ontology (Gangemi et al. 2002) and GeoSPARQL\(^7\). Further information concerning system ontology decisions can be found in Esswein et al. (2012).

**Components**

Components are run time elements of architecture that perform data transformation. A component can be an abstraction over a subsystem, each containing its

\(^7\) [http://www.opengeospatial.org/standards/geosparql](http://www.opengeospatial.org/standards/geosparql)
own set of architectural styles, components, connectors, and data elements. Subsystems are delineated into components such that each component serves a specific role with well-defined interface and connector semantics. Ideally, functionality is arranged among components so that coupling is minimized between components and cohesion is maximized within the components.

**Sensor System**

A Sensor System (SS) is an abstraction over a set of components that generate observation data. Commonly, this system includes a data acquisition device connected to a sensor or multiple sensors. A SS transmits measurements encoded as observations. A SS may optionally transmit metadata. From an architectural perspective, SSs must encompass a device with sufficient resources and connectivity to directly attach to the WQ CM. In cases where low power and network efficiency are at a premium, this device may be physically separate from the data acquisition device. A SS fulfills the role of Producer in the Worker Queue architectural style. Communication from a SS is push-based (one-way) for the purposes of this architecture. This does not preclude the use of bi-directional communications for other types of systems, e.g., sensor management service.

**Observation Agent**

An Observation Agent (OA) works on the behalf of a Sensor System to process and persist Observation data. Observation data is represented using RDF. The OA has three primary functions: 1) write observation to a Datastore component, 2) validate the syntax of the observation data, and 3) apply quality control checks. The OA is a
participant in multiple architectural styles. It assumes the Worker role in the WQ style, retrieving observations asynchronously from the WQ CM. It acts as Repository client to persist observations to the datastore (function #1). It acts as a REST client in order to achieve the second and third functions, which allows access to information beyond that which is included in the incoming WQ messages. Finally, it acts as a Publisher in the PS style to transmit observation data to one or more Subscribers.

**Metadata Agent**

A Metadata Agent (MA) works on behalf of a SS to store and provide access to metadata. Metadata is described using RDF. A MA can represent multiple SSs determined by their membership in a particular administrative domain. This domain is determined by the namespace of the Sensor System. MAs assume the roles of Queue Worker, Repository client, and REST Server. The Worker role retrieves messages from the CM. The Repository client persists metadata to a datastore. The REST Server provide read and write access to a RDF datastore using the SPARQL 1.1 Graph Store HTTP Protocol

**Observer**

An Observer is a consumer of streaming environmental sensing data. It adopts the roles of Subscriber and REST client. The Subscriber role receives fine-grained observation data, while the REST client role retrieves large-grained metadata on an as-needed basis. Observers encompass a broad range of functionality and may be used to support analysis, visualization, or archival of environmental sensing data.

---

**Metadata Authoring Tool**

The Metadata Authoring Tool (MAT) supports the creation and editing metadata. Metadata is represented using RDF and adopts the role of REST Client in order to communicate with the Metadata Agent.

**Architectural Viewpoints**

This section describes the system’s architectural viewpoints. Rozanski and Woods (2005) recommend using six viewpoints in an architectural description: 1) functional, 2) information, 3) concurrency, 4) development, 5) deployment, and 6) operational. This section introduces views based only on the Functional, Information, and Development viewpoints. Concurrency, deployment and operational viewpoints are not covered here. An informal description of the architecture from a deployment and operational viewpoint is available in Eidson et al. (2010) and Esswein et al. (2012).

**Functional Viewpoint**

This viewpoint approaches the architecture from a high-level, outlining the system’s overall structure and units of execution. Functional elements include components, data elements, and connectors. This viewpoint is useful as a basis for understanding the system’s functions. The information presented is beneficial for all stakeholders and required for users adopting the streaming data capability, e.g., low-level observers. Implementation specifics and jargon are minimized to benefit new or less technical stakeholders. This section can be considered a prerequisite for more detailed views presented from other viewpoints.
This viewpoint facilitates reasoning about the following system concerns: 1) modularity, 2) extensibility, 3) reusability, and 4) ease-of-use. Modularity is achieved by promoting a separation of concerns when grouping functionality into components. This viewpoint concisely describes the arrangement of components, allowing an evaluation of decomposition methods. Modularity leads to extensibility. Components are natural points of extension for new or replacement functionality, provided the semantics of the ports and connectors are maintained. Conversely, the adoption of functionality that cannot be captured in an existing component will entail a structural modification to the system. Individual components may be extracted and reused provided the use honors the semantics of the connectors. The semantics’ restriction is notable in this architecture because components tend to be specialized against multiple connectors. This specialization could result in reduced opportunities for reuse. The relevant connectors, ports, and components are depicted in a functional view, making the functional viewpoint an appropriate starting point for an evaluation of reuse.

Ease-of-use is a qualitative property of a system and can be difficult to isolate into a single viewpoint. The functional viewpoint provides the most reasonable starting point for evaluating the system’s ease-of-use. Architectural properties, like modularity, relate to ease-of-use. A system exhibiting poor modularity is more difficult to comprehend. Once stakeholders achieve a level of functional understanding, they can begin to focus attention on specific architectural components. For example, a field technician may be primarily interested in sensor system components while a scientist may limit their awareness to a particular observer component.
This viewpoint does not directly address concerns regarding performance, reliability, scalability, or security. These concerns fall in the domain of technically oriented stakeholders, e.g., operators, data owners, and developers. While this viewpoint is useful as a foundation for more technical viewpoints, its scope is intentionally limited to accommodate a broader range of stakeholders.

*Component and Connector View Conventions and Notation*

Component and connector (C&C) styles are used as the basis for the view shown in Figure 2-1. A C&C style describes a computational model that prescribes how execution, data, and control flow through a system (Clements et al. 2011, pg. 123). This view depicts the four specialized C&C styles identified in the Architectural Styles section. Figure 2-1 deviates from a conventional C&C view in its depiction of communication mediators and data elements. A communications mediator (CM) is typically abstracted as a connector element. Our C&C view treats a CM as a distinct entity in order to emphasize its role in facilitating interaction between two or more components. A CM is not appropriately represented as a component because it does not perform any data transformations. A second deviation from a conventional C&C view is the use of data as a first-class diagram element. In a conventional C&C view, data elements are implied by the combination of a component and a connector. The presence of multiple C&C styles (request/response, event-based, etc.) in our view leads to many possible component and connector combinations. To simplify interpretation of the model, data elements are explicitly specified as connector annotations. Annotations include
arrows to indicate the direction of data flow for a given data element. These arrows are not representative of control flow.

C&C View Discussion

The general data movement in Figure 2-1 flows from left-to-right. Observation data originates from the sensor system component. The multiplicity symbol shown below the SS indicates that more components exist then are shown by the diagram. Logically, all SSs are associated with two components: a MA and an OA. Operationally, a WQ CM disconnects the SS from the agents. When messages are queued by a SS, they are directed to either a general use observation queue or to a metadata queue specific to a MA namespace. A SS specifies the appropriate queue by passing a queue identifier parameter during the enqueue invocation. Both Agent types consume messages from their respective Queues and persist the messages to a Datastore. This is achieved using a Repository architectural style. Observation messages undergo a series of data transformations before being sent to the PS CM. OA data transformations are described in Esswein et al. (2012). The MA assumes responsibility for 1) updating the Repository with new metadata and 2) providing access to metadata stored in the Datastore. Metadata access is provided using the REST architectural style. In addition to metadata originating
Figure 2-1. Component and Connector View.
from SS, metadata may be added, replaced, or deleted using a REST client. The MAT is an example of a REST client that can create and manage metadata documents.

Observers are the final component in the data element’s life cycle. Observers act as Subscribers in the PS architectural style in order to receive observation traffic. If an Observer requires additional metadata to process an observation, it can interact with a MA using the REST style. This use of two architectural styles illustrates an important architectural design decision. Metadata usually changes less frequently than observation data and is more easily handled in large-grained documents as opposed to small-grained observation events. The REST Server component can support more sophisticated querying and filtering mechanisms than can be achieved with the topic-based message routing approach used by the PS architectural style.

Figure 2-1 offers several insights into the system concerns of modularity, extensibility, reusability and ease-of-use. The minimal dependencies of the SS and Observer components indicate a high degree of modularity. This is significant for this particular architecture since SS and Observers are anticipated to change on a regular basis. Similarly, the repositories involve a connection to one type of component, encouraging reuse and extension. A shortcoming of this diagram is a lack of interaction specifics regarding the connector between the datastore and the Agent components. Modularity in the Repository style is highly dependent on implementation specifics. For example, technology like the Open Database Connection\(^9\) (ODBC) driver can improve modularity compared with vendor-specific database connection methods. As evidenced

\(^9\) [http://support.microsoft.com/kb/110093](http://support.microsoft.com/kb/110093)
by the Figure 2-1, Observation and MA involve numerous connector attachments. This indicates that modularity and reuse is limited with these components.

**Information Viewpoint**

This viewpoint adopts a data-centric perspective of the architecture. It documents the structure and content of the information that passes through the system. The information content has significant influence over architectural design decisions. It is tied to the system’s concerns of modularity and scalability. Additionally, because the user’s ultimate goal is to access information, this viewpoint can provide information concerning ease of use. Operators and those observers requiring access to streaming data are the stakeholders most likely to benefit from this viewpoint.

**Ontology Modularization View**

Ontologies include a set of classes, constraints, and relationships. In this respect, ontologies are subject to the same decomposition principles as a system. Modularity in ontology descriptions reduces complexity by information hiding and encapsulation of functionality. Ontologies may be modularized to help description and data management. The SSN ontology documentation includes grouping classes into modules. From a programmatic standpoint, these groupings are useful for transferring larger-grained data elements. Additionally, analysis and metadata management tools can use modules as building blocks for information handling, e.g., translating them into objects or writing them to relational databases. The delineation of classes into modules can significantly impact software development. In order to evaluate the modularization used by this system, ontology coupling and cohesion metrics are applied. Oh and Ahn (2009) describe
a metric that characterizes the Number of Relation (NR) in a given module. This provides a measure of cohesion. It takes into account all relations and the distance of the relations:

\[
NR(M) = \sum_{i} \sum_{j} \frac{nr(c_i, c_j)}{|c_i|}
\]

\[
r(c_i, c_j) = \begin{cases} 
1 & \text{if link exist between } c_i \text{ and } c_j \\
\frac{1}{distance(c_i, c_j)} & \text{if link does not exist between } c_i \text{ and } c_j 
\end{cases}
\]

\[
distance(c_i, c_j) = \text{number of links between } c_i \text{ and } c_j.
\]

Where:

\[
M = \text{module} \\
c_i = \text{class, } c_i \in M \\
c_j = \text{class, } c_j \in M \\
|c| = \text{cardinality.}
\]

Oh and Yeom (2010) describe a second metric for module cohesion: the Hierarchical Relation (HR) metric. It is similar to the NR metric, but includes only hierarchical (subsuming) relations:

\[
HR(M) = \sum_{i} \sum_{j} \frac{hr(c_i, c_j)}{|c_i|}
\]

\[
hr(c_i, c_j) = \begin{cases} 
1 & \text{if hierarchical relation exists between } c_i \text{ and } c_j \\
\frac{1}{distance(c_i, c_j)} & \text{if hierarchical relation does not exist between } c_i \text{ and } c_j 
\end{cases}
\]

The Non-Hierarchical Relation (NHR) metric describes a counterpart to the HR metric (Oh and Yeom 2010). This relation is restricted to mereological relations:

\[
NHR(M) = \sum_{i} \sum_{j} \frac{nhr(c_i, c_j)}{|c_i|}
\]

\[
nhr(c_i, c_j) = \begin{cases} 
1 & \text{if hierarchical relation exists between } c_i \text{ and } c_j \\
\frac{1}{distance(c_i, c_j)} & \text{if hierarchical relation does not exist between } c_i \text{ and } c_j 
\end{cases}
\]
For module coupling, the Number of Separated Hierarchical Links (NSHL) and Number of Separated Relational Links (NSRL) metrics are used (Oh and Ahn 2009). These metrics describe the number of links, both hierarchical and relational, that exist between classes inside and outside of the module:

\[
NSHL(M) = \sum_i \sum_j nshl(c_i, c_j)
\]

\[
nshl(c_i, c_j) = \begin{cases} 
1, & \text{if hierarchical relation between } c_i \text{ and } c_j \text{ is disconnected} \\
0, & \text{otherwise}
\end{cases}
\]

\[
NSRL(M) = \sum_i \sum_j nsrl(c_i, c_j)
\]

\[
nsrl(c_i, c_j) = \begin{cases} 
1, & \text{if relation between } c_i \text{ and } c_j \text{ is disconnected} \\
0, & \text{otherwise}
\end{cases}
\]

All metrics were implemented using the Java\textsuperscript{10} programming environment and the Apache Jena\textsuperscript{11} Semantic Web development framework.

Metric results against the SSN ontology are shown in Table 2-2. Module delineation is based on the ontology documentation\textsuperscript{12}. By themselves, metrics offer little explanatory ability. The magnitude of the score is largely a product of the number of elements represented by the ontology. However, they are useful for comparison against alternative module decompositions. To illustrate how the coupling and cohesion metrics

\textsuperscript{10} http://java.sun.com
\textsuperscript{11} http://jena.apache.org
\textsuperscript{12} http://www.w3.org/2005/Incubator/ssn/ssnx/ssn
can guide module delineation decisions, an alternative grouping is described in Table 2-3.

This decomposition reflects a closer approximation to the conceptual model of environmental sensing concepts employed by this architecture. The modified decomposition has the same number of classes and relations as the original, but they are reorganized into fewer modules. Results from the metric calculations for the modified ontology are shown in Table 2-4. Cohesion metrics increased (modified – original = 1.69) while coupling metrics decreased (original – modified = 164). These findings indicate that the alternative decomposition exhibits improved modularity over the original decomposition.
Table 2-3. Modified Sensor Ontology Modules.

<table>
<thead>
<tr>
<th>Module</th>
<th>Constraint Block</th>
<th>Data</th>
<th>Deployment</th>
<th>Measurement</th>
<th>Capability</th>
<th>Measuring</th>
<th>Observation</th>
<th>Operating</th>
<th>Restriction</th>
<th>Platform Site</th>
<th>Process</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Output</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Observation Value</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Deployment</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Deployment Related Process</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Accuracy</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Detection Limit</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Drift</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Frequency</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Latency</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Measurement Capability</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Measurement Property</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Measurement Range</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Precision</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Resolution</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Response Time</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Selectivity</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sensor</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Property</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Stimulus</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sensor Input</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Feature of Interest</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sensing</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sensing Device</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Sensor Data Sheet</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Observation</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Maintenance Schedule</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Operating Property</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Operating Range</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Survival Property</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Survival Range</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>System Lifetime</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Battery Lifetime</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Operating Lifetime</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Operating Power Range</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Platform</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Input</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
</tr>
<tr>
<td>Output</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>Process</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>System</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
</tr>
<tr>
<td>Device</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>X</td>
</tr>
<tr>
<td>Condition</td>
<td>X</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

(41 Classes)

1 2 2 13 8 1 8 1 3 2

37
This viewpoint is from the perspectives of a system designer and developer. It is geared towards a technical audience of system operators, low- and mid-level observers, and data owners. The information covered is relevant to all system concerns. Three views are presented in this section. The tiered view and dependency structure matrix view are relevant to the concerns of modularity and extensibility. The module coupling view expands on the concepts introduced in the C&C view. It provides a graphical depiction of the locations and characteristics of dependencies between modules. This view serves as a supporting rationale for design decisions and is directly related to scalability and reliability concerns.

### Table 2-4. Modified Sensor Ontology Module Metrics.

<table>
<thead>
<tr>
<th>Module</th>
<th>Cohesion</th>
<th>Coupling</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Number Relations (NR)</td>
<td>Hierarchical Relations (HR)</td>
</tr>
<tr>
<td>Constraint Block</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Data</td>
<td>0.50</td>
<td>0.00</td>
</tr>
<tr>
<td>Deployment</td>
<td>0.50</td>
<td>0.50</td>
</tr>
<tr>
<td>Measuring Capability</td>
<td>0.88</td>
<td>0.85</td>
</tr>
<tr>
<td>Measuring</td>
<td>0.84</td>
<td>0.13</td>
</tr>
<tr>
<td>Observation</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Operating Restriction</td>
<td>0.63</td>
<td>0.50</td>
</tr>
<tr>
<td>Platform Site</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>Process</td>
<td>0.66</td>
<td>0.00</td>
</tr>
<tr>
<td>System</td>
<td>0.50</td>
<td>0.50</td>
</tr>
</tbody>
</table>

(14 modules)  4.51  2.47  2.04  316.00  6508.00

Totals:  9.02  6824.00
Tiered View

A tiered view is used to group components into logical layers or tiers. The tiers are represented as boxes, following the conventions of a layered architectural diagram described by Clements et al. (2011, pg. 96). Layered architectures impose restrictions on dependency relationships between components. Components in a tier may only interoperate with components in a neighboring tier. Arrows in the diagram indicate an allowed-to-use unidirectional relationship between components. Components adhering to a layered architecture are only allowed to use components in the same tier or in a lower tier. Cyclical allowed-to-use relationships are not possible. A tiered architecture allows reasoning about dependencies between components.

2-2 shows the Tiered View diagram. The tier classifications are drawn from the tiers defined by the CI: sensing fabric, backhaul and communications, middleware, and application. This system does not use the backhaul tier; hence, it is not included in the diagram. The sensing tier includes SS and appears on the lowest diagram tier. The middleware tier is positioned above the sensing tier and includes the MA, OA, and Datastore components. The third and final tier is the application tier, containing the Observer and MAT components. This view shows a straightforward depiction of the dependencies between components and conveys the rationale behind the arrangement of components. For example, Observer components are kept isolated from SS components. SSs have no dependency on other software components, simplifying configuration and deployment of the system.
A dependency structure matrix (DSM) is a tool for evaluating the correctness of dependency relationships in a layered architecture (Steward 1981). The DSM shown in Table 2-5 provides an alternative presentation of the view shown in Figure 2-2. Allowed-to-use relationships are identified between the components listed horizontally and the components listed vertically in the matrix view. In a layered architecture, no dependencies can exist above the diagonal of the matrix. The diagonal is indicated with an “X”. Positions with a hyphen indicate the absence of an allowed-to-use relationship. The DSM shown in the table confirms that the system meets the requirements of a layered architecture. The DSM approach is useful design aid when assessing the implications of future system modifications.

**Figure 2-2.** Tiered Architectural View.

**Dependency Structure Matrix View**

A dependency structure matrix (DSM) is a tool for evaluating the correctness of dependency relationships in a layered architecture (Steward 1981). The DSM shown in Table 2-5 provides an alternative presentation of the view shown in Figure 2-2. Allowed-to-use relationships are identified between the components listed horizontally and the components listed vertically in the matrix view. In a layered architecture, no dependencies can exist above the diagonal of the matrix. The diagonal is indicated with an “X”. Positions with a hyphen indicate the absence of an allowed-to-use relationship. The DSM shown in the table confirms that the system meets the requirements of a layered architecture. The DSM approach is useful design aid when assessing the implications of future system modifications.
Table 2-5. Dependency Structure Matrix.

<table>
<thead>
<tr>
<th>Component</th>
<th>Application</th>
<th>Middleware</th>
<th>Sensing Fabric</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uses?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>X</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MAT</td>
<td>-</td>
<td>X</td>
<td>-</td>
</tr>
<tr>
<td>OA</td>
<td>Yes</td>
<td>-</td>
<td>X</td>
</tr>
<tr>
<td>MA</td>
<td>Yes</td>
<td>Yes</td>
<td>X</td>
</tr>
<tr>
<td>D</td>
<td>-</td>
<td>Yes</td>
<td>X</td>
</tr>
<tr>
<td>SS</td>
<td>-</td>
<td>Yes</td>
<td>-</td>
</tr>
</tbody>
</table>

Module Coupling View

The view shown in Figure 2-3 documents the coupling dimensions of interface, data, synchronization, and identity. This view bears similarity to the Component and Connector view shown in Figure 2-1, both in structure and in notation. However, a number of distinctions exist. Notably, this view incorporates modules rather than components. The modules have a one-to-one correspondence with the components shown in the C&C view with the exception of the Datastore components. In our architecture, the Datastore introduces a single, one-way dependency: from an Agent to a Datastore. The details of this interaction are more appropriately described in an architectural description specific to a component. This view also incorporates data elements and interface specifics, which are typically excluded from a C&C view. This view supports reasoning about the system concerns of modularity and its derivatives: scalability, extensibility, and reusability. Additionally, this view aids in analysis of reliability from the standpoint of communications failures.
Conventions and Notation

The primary elements of this diagram are modules, CM, and connectors. Connectors are labeled with data elements and attached to ports. Like the C&C view, ports act as the junction between a connector and a component. Modules are labeled with one or more stereotypes that indicate a role in an architectural style. Modules that participate in multiple architectural styles will have multiple stereotypes. For clarity in description, modules are labeled by their specialization, e.g., Metadata Agent. CMs (e.g., WQ and PS) are shown with cloud symbols to distinguish them from components. Like components, CMs and connectors are joined using ports.

Interface notations are borrowed from the Unified Modeling Language\(^{13}\) (UML). UML lollipop and socket symbols are used in conjunction with a connector to illustrate an interface. The directionality of the lollipop and sockets indicates the path of control flow along a connector. These visual elements also serve to distinguish synchronous and asynchronous communication. Synchronous communications are shown with two lollipop/socket symbols in the standard convention of UML. This results in a total of four port connections for the interaction. UML does not provide a convention for representing asynchronous interactions. We use two conventions for asynchronous communications, depending on the nature of the interaction. The first method uses a single line with one lollipop/socket, resulting in a total of two port connections. The single lollipop/socket pair indicates that data movement occurs in only one direction (from socket to lollipop) and that communication is controlled (initiated) by the data source. The second

\(^{13}\) http://www.uml.org
asynchronous symbolization uses two lollipop/socket and two line symbols. This interaction includes three port connections and is distinguished from the synchronous symbology by connecting the second line to a socket symbol rather than a port. This form of asynchronous interaction relies on the data recipient to control (initiate) communication. The opposing direction of data-flow and control-flow is accomplished using a notification (callback) method, symbolized with a second socket/lollipop. Lollipop symbols are annotated (in italics) with their method label. Interface methods with parameters are shown with ellipses (“…”). This parameter abbreviation reduces clutters and avoids exposing implementation details. For illustration purposes, connectors are not required to explicitly link components. Disconnected lollipop/socket symbols can be associated based on their role in an architectural style. For example, the MAT Module has the role of REST client and is coupled to the REST server role of the MA.

View Discussion

The diagram in Figure 2-3 illustrates three architectural styles: WQ, PS, and REST. Starting from the left side of the diagram, a SS participates with Agents using the WQ style. This interaction involves a CM, as shown by a cloud in the figure. The SS issues an enqueue request to the WQ CM containing a message with an optional queue identifier. The enqueue operation provides a single uniform interface with well-known semantics. A message is directed to a CM queue based on its queue identifier. Messages without a queue identifier or with an “Observation” identifier are sent to a queue monitored by an OA. Otherwise, messages are sent, based on their queue identifier, to queues monitored by a MA. Communication between the SS and the queue is
asynchronous, as depicted by the single lollipop and socket interface in the figure. This implies that the execution of the SS is not blocked by the enqueue call, decoupling the interaction in the synchronization dimension. Note that the CM (and not the agent) may optionally issue an enqueue acknowledgement message. The CM allows an Agent to receive a message without being present when the message is transmitted, providing decoupling in the time dimension. The SS is required to maintain the identity of one WQ CM. In addition, the SS must be aware of its metadata queue name. The queue name convention is based on all or part of the SS namespace, e.g., www.intelligentriver.org/resource/subject1. Payload contents vary between the metadata and observation messages, but both employ the same RDF data model.

Agents are responsible for dequeueing messages from their respective queues. This is achieved through a notification mechanism. The notification mechanism is established during the dequeue(...) operation. Call parameters specify which queue is to be monitored. The interaction is asynchronous because the execution of the Agent is never blocked. The WQ CM is required to store the identity of participating Agents. This information is provided to the WQ CM by Agent components dynamically at runtime and is expected to change frequently over the course of the systems operation. This arrangement allows Agents to receive data from any number of SS components without having to maintain individual SS identities. Furthermore, it avoids the unreliability and performance penalties associated with synchronous calls. The enqueue, attach, and dequeue methods offer well established semantics, resulting in minimal interface coupling. All message payloads contain observation or metadata information encoded in
RDF. The link between the Agent and the SS is decoupled in terms of identity, synchronization, and time.

The OA and the Observer module participate in the PS architectural style. The Publisher stereotype associated with the OA module asynchronously sends messages to a PS CM. The PS CM directs messages to the appropriate Subscriber. The Publisher needs only to maintain the identity of the CM and is not responsible for tracking Subscribers, decoupling the interaction in identity. The PS CM can optionally employ queues to preserve observations, for a specified time period, in the absence of a subscriber client. This feature decouples the Publisher and the Subscriber in time. The publish interaction uses a single `publish(..)` method whose parameters specify the message and the topic-space to publish under. A Subscriber role asynchronously receives messages from the PS CM and only needs to be aware of the CMs identity. It uses a notification mechanism to receive messages without polling operations. This notification is established during the `subscribe` method call. All message payloads contain observation data described using the RDF data model. In summary, the Publish/Subscribe CM enables a many-to-many relationship between Publishers and Subscribers with decoupling in the time, synchronization and identity dimensions.

The Agents, Observers, and MAT depicted in the diagram participate in the REST architectural style. The MA acts as the HTTP server, providing client access to the metadata. The HTTP server is stateless and does not maintain identity information about its clients. It expects synchronous calls from clients, as shown by the two lollipop and
Figure 2-3. Module Interaction.
socket pairs. The REST style supports the use of a caching intermediary, a feature that can be utilized to remove time coupling for \textit{get(..)} operations. REST styles use uniform interfaces with well-known semantics, thereby reducing interface coupling. Responses have a content type of RDF. The OA, Observer and MAT modules act as clients in the REST style. REST clients require the identity of one or more REST servers. This information is determined dynamically at runtime. The runtime identity provision is made possible by the linked nature of the RDF data model, which provides the hypermedia functionality required of the REST style. The degree to which RDF satisfies the hypermedia constraint of REST is a subject of debate. RDF is a data interchange format that relies on uniform resource identifiers (URI) to link together data elements. Traditional hypermedia languages, like Hypertext Markup Language, also rely on (a subset) of URIs to direct the client towards other resource representations. However, unlike hypermedia, RDF data is not designed to control application state. This architecture makes an assumption that the Observer will trigger REST requests based on incoming Observation RDF messages. When the Observer attempts to dereference an RDF resource that is not present in its local RDF graph, it looks to a MA to provide the missing information.

**Conclusions**

This paper presents an architectural description of a streaming environmental sensing information system. This system provides the technological foundation for the four activities of data intensive science: 1) capture, 2) curation, 3) analysis, and 4) visualization. The provided architectural description defines the system’s terminology
and concepts in terms of architectural elements. Documentation and design rationale focuses on the structural properties of coupling and cohesion. Future work is needed to document viewpoints for security, concurrency, and testing.

Acknowledgements

This work is supported through the Clemson Public Service Activities Center for Watershed Excellence and the National Science Foundation (CNS-1126344).

References Cited


Gamma, Erich, Richard Helm, Ralph Johnson, and John Vlissades. 1995. Design Patterns: Elements of Reusable Object-Oriented Software. Boston, MA: Addison-Wesley.


Gray, Jim. 2007. “eScience: A Transformed Scientific Method.” In *Presentation to the Computer Science and Technology Board of the National Research Council*. Mountain View, CA.


CHAPTER THREE
TOWARDS ONTOLOGY-BASED DATA QUALITY VALIDATION IN LARGE-SCALE SENSOR NETWORKS

S. T. Esswein¹*, S. Goasguen², C. J. Post¹, D. White³, J.O. Hallstrom²,
G. W. Eidson⁴, and E.A. Mikhailova¹

¹School of Agricultural, Forest, and Environmental Sciences, Clemson University
²School of Computing, Clemson University
³Computing and Information Technology, Clemson University
⁴Department of Biological Sciences, Clemson University

Clemson, SC. 29634, USA

*Corresponding author. Current address: Department of Forestry and Natural Resources, Clemson University, 214 Lehotsky Hall, Clemson, SC 29634, USA. Tel. 540-272-6254, Fax: 864-656-3304.
Email address: sesswei@clemson.edu (S.T. Esswein).

Abstract

This paper presents an ontology-based approach for data quality inference on streaming observation data. The observation data originates from a large-scale sensor network deployed to support the Intelligent River® research initiative. Current methods for data quality evaluation are compared against ontology-based inference methods based on Semantic Web technologies. The quantity of monitoring locations and the frequency of data collection makes streaming processing a data intensive challenge. An event-based architectural style is employed to achieve streaming delivery of validated observation data.
data to multiple data consumers. Preliminary benchmark results indicate delays of 100ms for basic data quality checks based on an existing semantic web software framework. Results are maintained under increasing sensor data traffic rates by horizontally scaling data validation components. Results indicate that data quality inference using Semantic Web technology is possible with large-scale, data intensive sensor networks.

**Introduction**

Between 2006 and 2009, the southeast United States suffered a prolonged drought. The Savannah River Basin, a 27,500 square kilometre region consisting of portions of South Carolina, Georgia, and North Carolina, was particularly impacted. Unprecedentedly low river and reservoir water levels led to increased competition for water supply, hydropower, flood control, drought planning, recreation, water quality, fish and wildlife, and navigation (US Army Corps of Engineers 2011). The incongruity between water supply and water demand is expected to grow with expanding populations, industrialization and planetary climate change (Eidson et al. 2010). Accurate and timely access to environmental data is a critical component of the management solution. While hydrological and environmental data collection has long been an aspect of river management, increasing demands on water resources will require: (1) a broader spectrum of real-time data streams at ultra-dense temporal and spatial scales and (2) mechanisms capable of transforming, in real-time, these streams into actionable information suitable for informing river management strategies.

The real-time data stream approach has become a prevailing feature of sensor networks (Nittel 2009). Numerous middleware solutions have been developed to work
with data streams in data intensive applications (Wright et al. 2009; Aberer et al. 2007). Real-time data stream requires information management tools to process, analyse, visualize and model observation data (Balazinska et al. 2007). The translation of streaming data into actionable information has long motivated real-time monitoring. The United States Weather Service deployed networks of automated monitoring sites in the 1970’s to fuse real-time meteorological and hydrological data in order to detect and alert communities of impending flash flooding conditions (UCAR 2010). Automated environmental sensing has been growing steadily over the past few decades, with extraordinary growth in the past decade as networked, low-cost embedded sensing devices have become available. The quantity and variety of the data streams resulting from these sensor networks has created a data intensive processing challenge.

Sensor network technology is increasingly turning to ontology-based approaches for annotating, querying and reasoning about sensor data (Sheth et al. 2008). These capabilities have grown out of the Semantic Web vision described by Berners-Lee et al. (2001). The Semantic Web provides a common framework for sharing and reusing data across application, enterprise, and community boundaries (Herman et al. 2011). At the core of the Semantic Web is the Resource Description Framework\(^\text{14}\) (RDF), which provides a standard model for data interchange on the Web. The Web Ontology Language\(^\text{15}\) (OWL) builds on RDF to describe shared vocabularies of terms and relations. Ontologies for the Semantic Web are described in a format that machines can understand. A range of inference and reasoning tools exist that leverage the expressive capabilities of

\(^{14}\)http://www.w3.org/RDF/

\(^{15}\)http://www.w3.org/2001/sw/wiki/OWL
OWL and RDF. With regard to sensor networks Corcho and Garcia-Castro (2010) identify five areas where semantic technologies offer benefit: (1) data abstraction level, (2) data quality (and quality of service), (3) integration and fusion of data, identification and location of relevant sensor-based data sources, and (5) rapid deployment of applications.

This paper is interested in the contributions of semantic technology towards data quality analysis. The issue of expressing and quantifying data quality validation is a leading challenge facing sensing efforts (Sheth et al. 2008). This investigation evaluates data quality validation techniques applied to real-time data streams. The data originates from a large-scale sensor network supporting the Intelligent River® research initiative (Eidson et al. 2010). Semantic sensor network technologies are being explored as a means to improve streaming data processing and improve interoperability with the greater research community. The paper is organized as follows: Background and related work is described in Section III; Existing middleware and data quality methodology is described in Section IV; The semantic sensor network middleware and benchmark results are described in Section V; A discussion is provided in Section VI; Conclusions are drawn in Section VII.

Background and Related Work

Intelligent River Overview

The Intelligent River® observation instrument is a heterogeneous fabric of in situ sensors. The purpose of the instrument is to allow end-users, researchers, educators, and policymakers to collect, share and utilize a broad spectrum of hydrological and
environmental data at ultra-dense temporal and spatial scales. A high-level view of the instrument architecture is shown in Figure 3-1. Components are arranged into four tiers. The first tier implements a wireless sensing fabric involving aquatic and terrestrial sensing systems. The aquatic platforms incorporate stationary and profiling multi-parameter sondes to monitor rivers and lakes. The terrestrial platforms involve an array of instrumentation for monitoring a variety of parameters, e.g., groundwater, soil moisture, and rainfall. The second tier provides a transit and uplink system for relaying observation data from the sensing fabric to Clemson University’s high performance computing backbone. On the backbone, the third tier provides middleware for automating the validation and distributing observation data. The forth tier provides repository and presentation services for curating, analyzing and visualizing observation data.

Environmental sensor networks are increasingly relied on for scientific research, e.g. studies of water resource monitoring networks (Le Dinh et al. 2007; Guru et al. 2008), soil ecology (Szlavecz et al. 2007), volcanic activity monitoring (Werner-Allen et al. 2006), and light dynamics within shrub thickets (Selavo et al. 2007). Projects focused on sensing infrastructure with expressive metadata capabilities include Microsoft Research funded SenseWeb (Kansal et al. 2007; Luo et al. 2008) and the Swiss Experiment (Michel et al. 2009).
Open Geospatial Consortium Sensor Web Enablement

The Open Geospatial Consortium (OGC)\(^{16}\) provides a family of models, encodings and services to support environmental sensor webs. The Sensor Web Enablement (SWE) initiative enables interoperable and scalable service-oriented networks of heterogeneous sensor systems and client application (Reed et al. 2007). Interoperability focuses on syntactic compatibility based on shared XML schemas and Web service contracts. The SWE specifications relevant to this discussion include two languages: Observation & Measurements (O&M), Sensor Model Language (SensorML), and four services: Sensor Observation Service (SOS), Sensor Planning Service (SPS) Sensor Alert Service (SAS) and Web Notification Service (WNS). Planned additions to SWE include specification of a Sensor Event Service and Event Pattern Markup Language (Bröring et al. 2011). These additions will support a standardized approach to event stream processing.

\(^{16}\) http://www.opengeospatial.org
Semantic Sensor Network

Semantic technologies can improve interoperability and integration, as well as facilitate reasoning, classification and other types of assurance and automation (Lefort et al. 2011). Semantic interoperability supports high-level, context-sensitive information requests over heterogeneous information resources, hiding systems, syntax, and structural heterogeneity (Sheth 1999). Integration and fusion of data is aided by a data representation intended for machine reasoning.

The W3C Semantic Sensor Network Incubator Working Group (SSN-XG)\(^{17}\) developed an ontology for describing sensor network systems. The concepts and relationships of the SSN-XG ontology build from the Stimulus-Sensor-Observation ontology design pattern (Janowicz and Compton 2010). SSN-XG development included a survey of existing sensor network ontologies (Compton et al. 2009). Many of the ontology concepts draw from the existing Open Geospatial SWE standards, building on existing terminology and relationship information. The SWE standards primarily benefit syntactic and service interoperability, versus the semantic interoperability provided by the SSN-XG. The two technologies can complement one another by annotating existing SWE technologies semantic metadata (Sheth et al. 2008). Wider semantic web integration is facilitated by alignment with the DOLCE Ultra Lite (DUL) upper ontology (Gangemi 2002). The SSN-XG ontology syntax conforms to the OWL Description Logic (DL) sublanguage. The SSN-XG ontology is focused on the sensor network domain, avoiding domain specific concepts. Omissions include the description of network

\(^{17}\) [http://www.w3.org/2005/Incubator/ssn/](http://www.w3.org/2005/Incubator/ssn/)
configurations, spatial concepts, or measurement properties and units. In these cases, the SSN-XG can link to an external ontology, e.g. NASA’s Semantic Web for Earth and Environmental Terminology (Raskin and Pan 2005) or Climate and Forecast Metadata Conventions (CF)\(^{18}\).

Heterogeneous sensor systems and observation data are suited to ontology-based modelling. The SWE family of specifications provides a foundational model for describing sensor systems and observations, but is limited in the information it can express. Semantic web approaches offer a degree of extensibility not possible with the XML schemas used by the SWE. Additionally, reasoning and inference is possible with tools built based on Semantic Web technology. The advantages are highlighted by the challenges associated with expressing data quality and uncertainty. Challenges include variation in instrumentation types and data handling techniques, as well as a high incidence of sensing system errors and failures. Accurately characterizing data provenance and providing quality assurance is a critical concern for sensor network operators. A wide range of factors impact data quality, e.g., sensor design, platform location, calibration procedures, trust, weather conditions, and maintenance schedules. Data quality was a motivating use case during the development of the SSN-XG recommendation, resulting in the incorporation of data quality concepts into the model. For example, an individual sensor can provide measurement properties like drift, sensitivity, accuracy, measurement range, detection limits, latency, resolution, and precision. A system that powers collections of sensors can provide operational

\(^{18}\) http://cf-pcmdi.llnl.gov/
restrictions including battery life estimates. Provenance concepts are supported through vocabulary for description of installation, maintenance, etc. Other ontologies may be used in conjunction with the SSN-XG to formally describe data uncertainty. For example, the SWAP framework (Moodley and Tapamo 2011) uses the OntoBayes (Yang and Calmet 2005) ontology to enable Bayesian probability descriptions about sensor data.

Existing Middleware Architecture

Real-time Data Streaming

Central to the Intelligent River® cyberinfrastructure is a publish/subscribe architectural style. This event-based approach is common to distributed systems and cloud computing applications, e.g. OpenStack Compute\(^\text{19}\). It is suited to environmental sensing applications because of its ability to decouple the production and consumption of observation data. A subscribing software agent expresses its interest in a particular set of observations with a topic-based subscription pattern, rather than communicating directly with unpredictable observation producers. Architecturally speaking, Publish/Subscribe supports an unlimited number of streaming data Publishers and Subscribers. This avoids a key bottleneck in monolithic approaches involving a single streaming client or reliance on a data store intermediary.

Our middleware uses the open-source RabbitMQ\(^\text{20}\) implementation of the Advanced Message Queuing Protocol (AMQP) specification for publish/subscribe and queue based communications. Our original software implementation used the open

\(^{19}\) http://openstack.org/
\(^{20}\) http://www.rabbitmq.com/
source Narada Brokering\textsuperscript{21} message oriented middleware; migration to RabbitMQ improved performance and added additional clustering features. RabbitMQ provides a general-purpose messaging layer; messages are routed without regard to their payload. Client libraries are used to interact with the payload contents. Possible payloads relevant to sensor networks include: measurement/observation data, Public Key Infrastructure (PKI) certificates, metadata documents or control messages. This system uses Publish/Subscribe for small-grained observation and metadata messages. Larger-grained messages, like metadata documents and certificates, rely on RPC communications.

The topic-based subscriptions of AMQP provide a means of grouping and filtering observation data. Our convention is based around a hierarchy of concepts, e.g. organizations, projects, deployments, and platforms. Within this hierarchy, software agents can indicate an interest in a subset of observation data using a topic string.

**Streaming Data Representation**

Existing streaming data representation is done with a combination of two technologies. Early implementations relied on the Unidata Common Data Model (CDM) (Unidata 2008) to generate serializations of metadata and observation data. CDM describes a unified representation for multidimensional scientific datasets; supporting a range of model translations (e.g., to NetCDF, HDF, DAP). While powerful, the Unidata approach does not support computationally constrained gateway devices and can be cumbersome to process. To address these issues, metadata and observation data representation is supplemented with JSON. JSON is a lightweight data and metadata

\textsuperscript{21} http://www.naradabrokering.org/
message format. While JSON is limited in the information and relationships it can express, it is convenient for developers and resource-constrained devices.

**Sensing Fabric**

The sensing fabric is deployed with a purpose built sensing platform called a Motestack. The Motestack is a power efficient platform capable of interfacing with a variety of sensor formats common to environmental monitoring applications, e.g. analog, SDI12, 1-Wire, SPI etc. Communication is via 802.15.4/Zigbee radio, Wi-Fi or cellular. Motestack message size is minimized during observation transmission. A gateway device is used an intermediary between a low-power sensor network and the streaming data middleware. The gateway is a low-power embedded computer powered by larger batteries and/or solar power. The gateway translates binary Motestack messages to structured JSON messages. These messages are annotated with a subset of metadata describing a particular observation.

**Streaming Data Consumers**

Streaming data consumers are Subscribers in the Publish/Subscribe communications style. Sensor systems, by way of the middleware, send Subscribers a continuous sequence of observation messages. Messages are delivered to Subscribers using *implicit invocation* (Garlan and Shaw 1994), which removes the need for polling by the Subscriber. Once a message is published to the middleware, a Subscriber is expected to receive the message in a bounded time interval, i.e., real-time delivery. Publish/Subscribe allows any number of Publishers to broadcast observation data to any number of Subscribers. This allows the same observation data stream to be used
simultaneously for multiple purposes. A general-purpose Subscriber receives all observation traffic. General-purpose examples include data archival applications and real-time visualization tools. Different Subscribers may be employed to persist observations to different types of datastores, e.g. relational databases, comma separated value text files, and NetCDF binary files. A real-time visualization agent provides dynamic display of observation data and is useful for monitoring the operational status of the sensor network.

Subscribers can be used to perform validation checks on observation data. These checks apply heuristics to identify invalid observations based on completeness, threshold, and variation. Invalid observations are annotated with an error identifier and republished as a new observation. This approach ensures that both the original and the validated datasets are maintained. Other subscribing agents can choose to receive only validated observation based a topic-based subscription patterns. This data quality and assurance is used to assess and monitor the operation of sensing systems and network infrastructure. In the existing implementation, maintaining the data quality applications is made difficult by the limited expressive capabilities of our metadata. Configuration details, calibration instructions, and measurement capability rules cannot be stored in the current metadata description. Subsequently, this information is conveyed programmatically. The limitations of this approach become more pronounced as sensor systems are added. Additionally, the current approach offers no guarantee that republished observations with data quality annotations will conform to the real-time requirements available with non-annotated observation data.
Conventional Data Consumers

Conventional data consumers do not require streaming access to observation streams. Data access is achieved with request/response interactions. Data access applications include web portals and the OGC SWE Web Services. The web portal provides limited capabilities for data access through a simple web browser interface and is primarily intended for use by the general public. The SWE Web Services offer more powerful query mechanisms for accessing observation data and metadata. The SWE Sensor Observation Service uses a relational database that is populated with observation data from a streaming data subscriber. Support for other data distribution mechanisms are planned, e.g. CUAHSI Hydrologic Information Service (HIS).

Semantic Sensor Network Middleware

Semantic Web technologies offer data representation advantages over the existing middleware. The expressivity of RDF data formats provides superior flexibility and extensibility. Ontology-based vocabularies and reasoning provides data consistency and the expression of sophisticated rules. While Semantic Web technologies offers many benefits, its is not clear how well semantic inference tools perform data intensive settings. This section explores the applicability of adapting semantic sensor network technologies into a streaming observation data approach. A semantic inference software agent is developed to processes incoming observations before they are published to a publish/subscribe broker. The semantic inference agent serves three purposes: (1) ensure that the observation message is well formed RDF, (2) validate against an ontology

http://his.cuahsi.org/
expressed with OWL and (3) perform data quality checks on the measurement data using custom rules. Purpose three is analogous to the data quality software agents described in section III. Once validated and annotated, observation data is published to an AMQP publish/subscribe exchange and distributed to Subscribers.

The SSN-XG ontology is used as to model the components and processes involved in acquiring and processing observation data. The SWEET ontology is used as a vocabulary for describing specific environmental phenomena. The OGC GeoSPARQL\textsuperscript{23} ontology is used for representing geospatial locations. For ontology modelling and editing we use the open source Protége\textsuperscript{24} tool developed at Stanford University.

The Jena API\textsuperscript{25} provides a software framework for semantic data manipulation and reasoning. Jena simplifies working with RDF graphs and ontologies. The Jena rule-based inference engine is the central component of our observation data validation and data quality checks. The Jena internal inference engine supports the syntax of the three OWL variants, distinguished by their expressivity: OWL Lite, OWL DL, and OWL Full. Jena provides varying degrees of reasoning support. The performance of an application using reasoning is highly dependent on the configuration of inference and reasoning engine. In addition to its internal inference engine, Jena can utilize an external reasoner provided it adheres to the DIG interface standard. We experimented with the three internal reasoner configurations implementations and found the Jena OWLMicro and OWLMini configuration options offered enough performance while maintaining

\textsuperscript{23} http://www.opengeospatial.org/standards/geosparql\textsuperscript{24} http://protege.stanford.edu/\textsuperscript{25} http://incubator.apache.org/jena/
sufficient inference capabilities to allow the required queries. In addition to OWL support, Jena supports custom defined rule sets. These rules can be used to perform specialized data validation checks (Calder et al. 2010).

Sensing devices in our test environment our software emulators to allow for easy adjustment of publish rates and provide greater consistency in network latencies. These devices send observations to an AMQP queue prior to their publication on an AMQP topic exchange (publish/subscribe). This differs from our existing middleware approach, but ensures that observation data is correctly validated and measurements have data quality annotations before they are published on the topic exchange. A diagram depicting our semantic sensor network architecture is shown in Figure 3-2. The TURTLE RDF serialization format is used in place of RDF/XML due to its concise representation and human readability. Observation data is kept separate from the sensor metadata to keep message sizes reasonable (e.g. 4KB for a message containing 3 observations). RDF observations are published to a RabbitMQ topic exchanges through a specialized gateway application.

**Semantic Reasoner Agent**

The semantic reasoner software agent implements the OWL validation and custom data quality checks described above. Upon start up, the SSN-XG ontology is loaded into a Jena model, which is an in-memory graph representation. Sensor System metadata, expressed using the SSN-XG, is then loaded into the same model. The metadata contains information describing sensors, measurement capabilities, properties etc. Measurement capabilities include concepts relevant to data validation procedures,
e.g., a measurement range. A measurement range is specific to a sensor type is typically provided by the sensor manufacturer in a datasheet. Measurement capabilities associated with a sensor may change under varying sensing conditions. The reasoning agent may optionally load a set of custom rules specified in a configuration file. These rules are parsed into a `GenericRuleReasoner` instance in preparation for incorporation into an Inference-enabled model. The reasoner agent then begins to listen for messages from an AMQP Queue. When the reasoner agent software receives a message, the observation is parsed and loaded into an inference model. Once loaded, OWL validation and custom rule checks are performed. The results of the checks are annotated into the observation message. The agent then publishes the annotated observation to an AMQP publish/subscribe topic exchange. Jena provides its own syntax for the rules that are fed into a `GenericRuleReasoner`. The general format involves a series of premise terms and a series of conclusion terms. The following is an example of a forward-chaining measurement range check rule that annotates an observation with a valid flag if the range check is valid:
Figure 3-2. Detailed System Diagram.
The Jena inference engine presents a significant bottleneck when incorporated directly into a streaming data approach. Transit delays on our existing publish/subscribe middleware generally fall between 10-20ms depending on network configuration. The delays incurred by the semantic reasoner software agent are much higher. This is particularly evident when rates of observation publishing increase. In data intensive scenarios, the queue in front of the reasoner agent grows much faster than the agent can empty it. To address this, we allow multiple reasoner agents to connect to a single queue. This allows the inference checks to be carried out in parallel. Adding or removing reasoner agents requires only a modification to the worker pool and can be adjusted to match the current needs of the system. Although not implemented for this test, it is possible to monitor the size of the task queue and automatically provision reasoner agent resources according to current system conditions.
Benchmark Results

A series of benchmark tests were performed to determine the observation delay penalties associated with different reasoner agent configurations against varying observation-publishing rates. This description includes results using the OWL Micro model validation and the simple measurement range check described above. Delays are calculated based on subtracting the time at which an observation is received at the AMQP subscriber from the time the measurement was published. This represents transit delay measurement. Publish rates are adjusted by a software application that emulates a sensing device. The emulator publishes batches of observations at a specified rate. Our tests publish 25 messages in a single batch. The task queue is allowed to clear before the next publish rate batch is transmitted. Average transit delay results are shown in Figure 3-3. The influence of batch size on transit delays are shown in Figure 3-4. Four identical tests are run with varying numbers of reasoner agents listening on the task queue. The zero agent time series represents the delay when inference checks are disabled, and messages are forwarded directly to the Publish/Subscribe topic exchange.

Jena performance is highly dependent on configuration settings, resulting in highly variable benchmark results. The time required for the reasoning and rule checking is also dependent on the observations. SSN-XG observations may vary in size and message content. To allow accurate comparison between results, a single layout was used for all observations. The goal of this benchmark was to evaluate semantic reasoning times relative to various configurations and agent configurations. Optimization of the
RabbitMQ middleware was not considered. Tests were carried out using Pentium 4 workstations with 4GB RAM and gigabit Ethernet connectivity.

The OWL Micro validation and the measurement range check took approximately 100ms to complete. This threshold is evident in Figure 3-3. When a single reasoner is used, the performance declines quickly near a publish rate of 10 observations per second. At this point, the queue grows very quickly as the reasoner agent is unable to keep up with incoming messages. If the observation batch size is increased from 25, the backlog becomes more pronounced (See Figure 3-4). Adding reasoner software agents increases the threshold at which transit delays increase. Figure 3-3 shows the addition of agents and corresponding reduction in transit delays.

Discussion

Semantic Web technology offers many benefits for environmental sensing systems. The combination of RDF, RDFS and OWL offer an expressive language for describing sensor systems and sensing processes. This capability allows a wider variety of devices to be described than approaches employed by the existing metadata middleware. The use of a common ontology allows sharing of meta- and observation data with a larger research community. Semantic Web reasoning tools offer opportunities for data validation methods based on portable rule sets versus compiled program logic. This allows extensibility without re-compilation or program modification. As Semantic Web adoption grows, it offers exciting opportunities for data fusion with data sources obtained through other environmental sensing implementations.
Figure 3-3. Average Transit Delay Benchmark Results.
Figure 3-4. Batch Size Benchmark Results.
Adaptation of existing metadata into the SSN-XG ontology proved challenging due to unfamiliarity with the underlying technologies and absence of high-level tools for metadata authoring and editing. Metadata was converted from existing representations programmatically using the Jena framework. Conceptually, mapping from the existing metadata was simplified by the correspondence between the SSN-XG ontology and the OGC SWE terminology. The SSN-XG ontology is organized into modules, which partition the classes into high-level groupings. The modules group similar functionality and simplify metadata authoring. Rather then modeling an entire sensor network design at once, modeling can be focused on a particular area of the overall system. The SWEET ontology played a minor role in our test implementation. It provides a comprehensive classification of terminology associated with environmental sensing. The wide coverage of concepts has benefits and limitations. Its generality supports a wide audience of potential users, but comes at a cost of reduced relevance to our particular application domain. A domain specific ontology, e.g., water resource monitoring, leads to a straightforward mapping to the environmental properties sensed by our sensing systems. Benchmark results indicate that latency associated with ontology and rule validation is low enough to accommodate streaming usage. Additionally, results show that transit delays can be maintained under increasing traffic rates by increasing the number of reasoning agents.

Future implementations of the reasoner agent are anticipated to build on the simple measurement range checks to provide more sophisticated sequential data analysis techniques, e.g., cumulative sum calculations. This functionality is performed in the
current middleware, but suffers from inflexibility and low tolerance to sensor failures, e.g., moving window may incorrectly flag an observation as out-of-bounds if a gap appears in a data stream.

**Conclusions and Future Work**

This paper evaluates the suitability of adopting an ontology-based approach to metadata description and semantic reasoning with the Intelligent River® monitoring middleware. This middleware supports data intensive monitoring with large numbers of sensing systems and types of sensing devices. Data is passed through the middleware and into streaming data subscribers who then analyze or visualize the data into information that can be used to further river research activities or inform decision makers.

Future work involves exploring other ontology and semantic tool options. For example, the implementation described in this paper relied on flat file serializations of RDF data. A more robust approach, suitable for multi-user access, is needed for a production implementation. Future work will include the incorporation of a RDF-specific triplestore approach to multi-user data access. Another area requiring further attention is the authoring, editing, and management of metadata. Furthermore, working directly with RDF may be difficult for those responsible for creating and managing metadata. Work is under way to develop a simplified user interface for interacting with sensor network RDF data.

There is a trend towards pushing observation data processing out to the edge of a sensor network. This is made possible with higher power embedded computing and allows the sensing fabric to act more intelligently, supporting data fusion or actively
responding to events. Supporting ontology-based inference capabilities on the sensing platforms would provide a powerful extension of the approach described here. Sensing devices could be programmed with rules guiding their activities and allow devices to act with greater autonomy. However, this would require significantly more resources than those currently available with most low-power mote-class devices. An alternative would be to provide these capabilities on an intermediate gateway device, deployed with the low-power devices but supported by greater power and computational resources.

Acknowledgements

This work is supported through the Clemson Public Service Activities Center for Watershed Excellence and the National Science Foundation (CNS-1126344).

References Cited


CHAPTER FOUR

BIOINFILTRATION BASIN MONITORING, MODELING AND VISUALIZATION USING GEOGRAPHIC INFORMATION SYSTEMS IN AIKEN, SC

S. T. Esswein1*, C. J. Post1, E. A. Mikhailova1, G. W. Eidson2, and J.O. Hallstrom3

1School of Agricultural, Forest, and Environmental Sciences, Clemson University

2Department of Biological Sciences, Clemson University

3School of Computing, Clemson University

Clemson, SC, 29634, USA

*Corresponding author. Current address: Department of Forestry and Natural Resources, Clemson University, 214 Lehotsky Hall, Clemson, SC, 29634, USA. Tel. 540-272-6254, Fax: 864-656-3304.

Email address: sesswei@clemson.edu (S.T. Esswein).

Abstract

Low Impact Development (LID) philosophy is increasingly guiding stormwater management practices. Bioinfiltration technologies are among the most widely applied LID management technique in urban watersheds. The benefits of bioinfiltration systems include reductions in peak flows, increases in ground water recharge, decreases in runoff volumes, pollutant filtering. These systems are a relatively new technology and evaluations of long-term performance are limited. This study describes a bioinfiltration monitoring program occurring over the period of Spring 2011 to Summer 2012 in the City of Aiken, SC, USA. Monitoring data were used to 1) evaluate changes in bioinfiltration performance over time, and 2) parameterize a bioinfiltration model.
Analysis includes a characterization of soil physical and hydraulic characteristics for the study region. Spatial data analysis found a lack of spatial autocorrelation among observed infiltration rates. A novel analysis technique detected an effective porosity reduction of 1.54 m$^3$ m$^{-3}$ in the bioinfiltration media over the study period of 1-year. This finding points to reductions in storm water handling capacity over time. Further study is necessary to determine whether this reduction is associated with post construction settling of the media or with longer-term trends in media composition. The implications of soil property spatial structure and media property temporal structure are evaluated using a grid-based infiltration and rainfall excess model. The model is implemented in a Geographic Information System (GIS) framework and supports parameterization at within-catchment spatial scales. Model results are illustrated using 3D visualization techniques.

**Keywords:** bioinfiltration, geographic information systems, GIS, infiltration modeling, low impact development, rainfall runoff modeling, stormwater, urban soils

**Introduction**

The hydrology of urban areas is influenced by high percentages of impervious cover and alterations to soil and vegetation (Arnold et al. 1996; Stankowski 1972; Pitt et al. 2000). These factors lead to increased frequency and magnitude of stormwater production, degradation of water quality, and alterations to ground water recharge (US EPA 2007). Management decisions can reduce the negative impacts of stormwater at the local and watershed scale. Stormwater management has traditionally focused on conveyance of storm water away from urban centers as quickly as possible.
(Niemczynowicz 1999; Arnold et al. 1996). Recently, Low Impact Development (LID) design practices have shown promise as an alternative to traditional storm water control structures. Low impact development practices emphasize local or in-situ controls, addressing rainfall excess closer to its source and have been shown to reduce storm-water volumes, reduce peak flows, provide pollutant filtering (Endreny 2004; Hatt et al. 2009; Kim et al. 1999). Examples of LID in-situ methods include porous pavement, bioswales, and bioinfiltration systems, with bioinfiltration basins being the most widely adopted (Davis et al. 2009).

A bioinfiltration Best Management Practice (BMP) is a bowl shaped depression over a constructed permeable subsoil (Heasom et al. 2006). The efficiency of water movement into and through the permeable subsoil or media determines the effectiveness of the BMP (Thompson et al. 2008). The hydraulic properties of the media are engineered to exhibit infiltration performance characteristics based on a design specification. For example, a bioinfiltration system designed for pollutant filtering will incorporate lower infiltration rates compared to a system designed to maximize storage capacity. The depth of the media may vary, with typical depths between 0.7 – 1 m deep (Davis et al. 2009). Surface depression depth and volume is dictated by designed volume requirements and available space, ponding depths typically range from 15-30 cm (Davis et al. 2009). Media depth decisions may also be influenced by vegetation choices. Engineered media is not suited to certain types of vegetation and root systems may require contact with native soil for survival. Ponding volume exceeding the basin capacity is conveyed to conventional stormwater structures. Exfiltration from the engineered media is governed by the
properties of the native soil beneath the excavated area. Soil properties are subject to alterations resulting from excavation techniques and compaction by heavy machinery (Brown and Hunt 2010). Designs may incorporate an under drain component to allow greater control over exfiltration rates. Depending on the under drain configuration, a bioinfiltration basin may act as a detainment control rather than a true retention control. The latter case may contribute a portion of processed stormwater volume to downstream stormwater controls.

Bioinfiltration systems are a relatively recent LID practice and research evaluating their long-term performance and maintenance requirements is limited (Davis et al. 2009). Studies investigating trends in multi-year performance of bioinfiltration sites have produced varying results. Lindsey et al. (1992) found that 27% of infiltration basin BMPs surveyed functioned as designed after four years, with 46% of these failures attributable to “excessive sediment or debris”. Excessive sediment washes into the BMP and can become trapped in the void spaces of the surface layer. However, Jenkins et al. (2010) found that sediment deposits were not an issue after eight years. Similarly, Emerson and Traver (2009) did not find any systematic reductions in performance over a four-year period. Besides sedimentation, mechanical compaction, raindrop impact, repeated ponding influence long-term performance of infiltration BMPs (Brown and Hunt 2010; Davis et al. 2009; Pitt et al. 2000). Thompson et al. (2008) found that declines in infiltration due to compaction were dependent on the composition of engineered media.

The uncertainty associated with the design and lifecycle of bioinfiltration BMPs makes monitoring and maintenance a crucial component of their success. Asleson et al.
(2009) suggest three alternatives for evaluation of bioinfiltration BMPs: (1) visual inspection, (2) infiltration rate testing, and (3) synthetic drawdown testing. These methods involve periodic visits to the bioinfiltration site. Continuous monitoring using automated methods has been employed for long-term evaluation (Brown et al. 2012) and may yield information not otherwise available. For example, a multi-year continuous monitoring survey conducted by Emerson and Traver (2009) found that hydraulic conductivity has a seasonal dependence. Common continuous measurement parameters for bioinfiltration systems include inflow and outflow measurements, ponding level, local precipitation, and soil water content. Wireless sensor network (WSN) technology is appropriate for continuous monitoring of soil water content at higher resolutions or greater spatiotemporal scales (Bogena et al. 2010). A WSN uses low-cost, power efficient sensing devices capable of streaming observation data to remotely located data consumers. Streaming data can be processed and validated using automated techniques. This simplifies data management and avoids the “information overload” potential that exists with dense deployments of data acquisition devices. Other advantages include the ability to automatically identify faults and notify responsible parties when failures occur (Eidson et al. 2010b).

The optimal design of bioinfiltration systems is complicated by the challenges associated with characterizing surface hydrology in urban areas. These challenges stem from alteration to topography, additions of impervious surfaces, and routing of stormwater flows to underground conveyance systems. Urban surface hydrology modeling methods may be soil physical theory based (e.g., Richard’s equation, Green-
Ampt equation) or empirically based (e.g., Soil Conservation Service Curve Number, Rational Method). Various forms of both methods have been applied to model bioinfiltration systems (Browne et al. 2008; Dussaillant et al. 2004; Heasom et al. 2006) and are widely applied in urban stormwater design tools, e.g., U.S. EPA’s Storm Water Management Model (SWMM)\textsuperscript{26} and the U.S. Army Corp of Engineer’s HEC-HMS\textsuperscript{27}. Parameterization and calibration of infiltration models requires detailed precipitation, topography, and soil property information. Geographic Information Systems (GIS) can aid in data management, data preparation, and analysis of model results. Grid modeling approaches, based on cell-based raster data, can perform prediction at a within-catchment spatial scale. Within-catchment scales describe a model that discretizes space into elements smaller than a hydrologic catchment. This scale is necessary to capture the localized hydrological processes that govern bioinfiltration performance and cannot be adequately described by point or lumped approaches. Spatially explicit grid-based models can be difficult to apply. Parameter values are required for each grid element, information that may be costly and difficult to obtain or accurately estimate (Beven 2001). Furthermore, the resolution and extent of grid-based models is constrained by available computational resources.

Advancements in computing and geospatial data acquisition technology support increasingly data-intensive modeling scenarios. Light Detection and Ranging (LIDAR), a remote sensing technology, has become the predominant method of obtaining high-resolution terrain models available over large areas (Liu 2008). Ground penetrating radar

\textsuperscript{26} http://www.epa.gov/athens/wwqtsc/html/swmm.html
\textsuperscript{27} http://www.hec.usace.army.mil/software/hec-hms/
(GPR), another remote sensing technology, provides spatially continuous measurements of soil moisture. While promising, data products like GPR present challenges of their own. Huisman (2003) found that GPR use is hampered by the complexity associated with acquiring and processing the data. Traditional in-situ observation methods for soil moisture data and infiltration rates may become cost-prohibitive or impractical for data collection over large areas or at fine scales. Numerous studies have shown that soil infiltration rates exhibit limited spatial dependency, even at local scales (Beven 2001; Greminger et al. 1985; Sobieraj et al. 2004). This restricts the applicability of estimation or interpolation techniques. At catchment or regional scales, soil map unit and series may adequately describe spatial heterogeneity of infiltration. However, at local scales biological processes such as tree roots, earthworm burrows (Lee 1985) or ant nests (Eldridge 1994) may dominant. Study scale determines estimation techniques. In other words, “processes and parameters important at one scale may not be as important or predictive at another scale” (Turner 1989). Estimation of soil properties in urban areas presents an additional challenge, as urban soils are subject to anthropogenic disturbance, e.g., compaction, destruction of original soil horizons (Pitt et al. 2000). Uncertainty associated with soil property estimation has implications for bioinfiltration system design. Variations in soil properties greatly influence the infiltration rates of soil and media (Kale 2011). Sensitivity analysis performed with the Green-Ampt infiltration model shows porosity and hydraulic conductivity as the most influential model parameters (Skaggs and Khaleel 1982). Despite the limitations, infiltration models are valuable design tool for stormwater engineers, supporting estimation of pre- and post-
construction bioinfiltration performance. Grid-based infiltration models can describe and illustrate infiltration processes at within catchment spatial scales. This spatial scale is important when modeling bioinfiltration systems that may only drain contributing areas encompassing a few city blocks. Grid-based approaches are also conducive to visualization, which opens up the decision making process to a non-engineering audience.

This paper evaluates bioinfiltration systems installed as part of a green infrastructure initiative in the City of Aiken, S.C., which is located in the southeastern United States. Monitoring data is used to evaluate the efficacy of LID practices and identify temporal trends in media properties. This research focuses on within-catchment scales, using soil, topography, and hydrologic data encoded as cell-based raster data. Analysis and modeling is performed within a GIS framework and illustrated using 3D visualization techniques. The objectives of this research were: 1) summarize the physical and hydraulic properties of soil within the City of Aiken, SC; 2) evaluate the spatial structure of saturated hydraulic conductivity within the study area; 3) examine temporal trends in effective porosity and evaluate its potential as a bioinfiltration performance indicator; 4) implement a grid-based infiltration and rainfall excess model geared towards modeling bioinfiltration systems at within catchment scales; 5) visualize modeled bioinfiltration basin performance.
Materials and Methods

Study Area

The bioinfiltration basins under observation are located in the City of Aiken, South Carolina (33.549397° N, -81.720689° W), which is located in the Upper Coastal Plain physiographic region of the southeastern United States (Figure 4-1). Monitoring and soil characterization is performed on a 0.62 square kilometers (0.24 sq. mi.) area of interest (AOI) located in the central commercial district (Figure 4-2). The County of Aiken receives an average of 117.73 cm (46.35 in.) of precipitation annually, with a 30-year temperature normal between 11.17 to 23.89 degrees Celsius (52.1 - 75.0 F) (SC DNR 2011). The AOI is predominantly Orangeburg loamy sand (OrA, 0.43 sq. km² or 69% of AOI) soil with some Fuquay sand (FuB, 0.191 sq. km² or 31% of AOI) soil (Figure 4-2 and Table 4-1). Fuquay and Orangeburg soils belong to hydrologic soil group B. Group B soils consist of moderately deep or deep, moderately well drained or well drained soils with moderately fine to moderately course texture. Group B soils have a moderate infiltration rate when thoroughly wet.

The City of Aiken lies on the boundary of the Savannah (HUC 030601) and Edisto (HUC 030502) river basins, with the majority of surface water draining towards the Savannah River by way of the Sand River. The general topography of the Upper Coastal Plain is low relief. However, relatively wide ranges of elevations exist within the city boundary. A LIDAR derived elevation models show an elevation range between

---

28 Data available from the U.S. Geological Survey
Figure 4-1. Map of study area, monitoring sites, and bioinfiltration basins.
Figure 4-2. Map of soil sample sites and soil map unit boundaries.
Table 4-1. Selected soil physical and chemical properties for Fuquay sand (FuB) and Orangeburg loamy sand (OrA) for Figure 1 (Source: USDA/NRCS Soil Data Mart, 2012).

<table>
<thead>
<tr>
<th>Horizon(s)</th>
<th>Depth (cm)</th>
<th>Soil pH</th>
<th>Organic matter (%)</th>
<th>Clay</th>
<th>Moist bulk density (g cm(^{-3}))</th>
<th>Saturated hydraulic conductivity (mm hr(^{-1}))</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fuquay sand (FuB), 2-6 % slope</strong> (loamy, kaolinitic, thermic Arenic Plinthic Kandiudults)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ap</td>
<td>0-20</td>
<td>4.5-6.0</td>
<td>0.5-2.0</td>
<td>1-7</td>
<td>1.60-1.70</td>
<td>151.20 – 507.60</td>
</tr>
<tr>
<td>E</td>
<td>20-53</td>
<td>4.0-6.0</td>
<td>0.0-0.5</td>
<td>0-15</td>
<td>1.45-1.55</td>
<td>151.20 – 507.60</td>
</tr>
<tr>
<td>E</td>
<td>53-88</td>
<td>4.5-6.0</td>
<td>0.0-0.5</td>
<td>10-35</td>
<td>1.40-1.60</td>
<td>14.40 – 50.40</td>
</tr>
<tr>
<td>Bt1, Bt2, Btv1, Btv2</td>
<td>88-175</td>
<td>4.5-6.0</td>
<td>0.0-0.5</td>
<td>20-35</td>
<td>1.40-1.60</td>
<td>1.51 – 5.04</td>
</tr>
<tr>
<td><strong>Orangeburg loamy sand (OrA), 0-2 % slope</strong> (fine-loamy, kaolinitic, thermic Typic Kandiudults)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ap, BA</td>
<td>0-15</td>
<td>4.5-6.00</td>
<td>0.5-1.0</td>
<td>4-10</td>
<td>1.35-1.55</td>
<td>50.40 – 151.2</td>
</tr>
<tr>
<td>Bt1</td>
<td>15-60</td>
<td>4.5-5.5</td>
<td>0.0-0.5</td>
<td>18-35</td>
<td>1.60-1.75</td>
<td>14.0 – 50.40</td>
</tr>
<tr>
<td>Bt2</td>
<td>60-150</td>
<td>4.5-5.5</td>
<td>0.0-0.5</td>
<td>18-45</td>
<td>1.60-1.75</td>
<td>14.0 – 50.40</td>
</tr>
</tbody>
</table>
87.48 and 168.76 meters (287.01 – 553.68 ft.). The low minimum elevations correspond to channels of the Sand River, which has canyon-like morphology in its headwater reaches. The highly incised channels are thought to have formed within the last 100 years in response to increased runoff stemming from urbanization-induced alterations in hydrologic response (Julian and Torres 2006).

As part of an ongoing effort to address stormwater impacts to the Sand River, the City of Aiken has undergone a series of green infrastructure improvements, including the installation of nine bioinfiltration basins and placement of pervious pavement in parallel parking spaces and parking lots (Eidson et al. 2010a). All basins are located in wide (approximately 28 meter) vegetated medians located between roadways (Figure 4-3). A map of bioinfiltration basins is shown in Figure 4-1. Abbreviations of basin locations are based on adjacent roads. An abbreviation key is provided in Figure 4-1. The median areas (cells) containing bioinfiltration basins are separated from roadways by raised curbing and covered with grass or landscaped vegetation. Vegetation includes shrubs, grasses, and mature trees. Engineering design and construction was done by Woolpert, Inc. with input from the City of Aiken and faculty from Clemson University. Each cell configuration varies by depth, volume, vegetation, soil media, under drain presence, and connection to storm-water controls. Selected cells were connected to larger drainage areas by way of configurable storm-water sewer inflows and curb cuts. Five bioinfiltration cells (abbreviated BRC) were chosen for intensive monitoring (Figure 4-1).
Native Soil Sampling

A baseline inventory of soils was performed using the Web Soil Survey and the SSURGO dataset for the AOI. Soil survey data includes soil order, texture, and saturated hydraulic conductivity (Table 4-2). Native soil borings and laboratory analysis were performed prior to construction. A total of 20 test borings were collected and analyzed by an independent engineering consultant (Fairbanks and Wargo 2009). Ten of these borings were obtained from potential BRC sites. These borings were taken to a depth of approximately 1.83 meters (6 ft.) below grade using a direct push method\textsuperscript{29}. The remaining ten borings were drawn from areas with pavement cover to variable depths using a hollow stem auger. Selected samples were evaluated in a laboratory for natural moisture content and gradation analysis (Table 4-2).

Infiltration tests were performed in the ten vegetated median sites within twenty-four hours of well excavation and described in the technical report of Fairbanks and Wargo (2009). A polyvinyl chloride (PVC) casing with the lower 0.91 meters (3 ft.) screened and slotted was placed in the borehole. Boreholes were repeatedly filled with water over a twenty-four hour period to achieve saturated conditions. The infiltration test was performed by filling the casing with water and monitoring the change in water level over time. Level and time was recorded until the well completely drained or a stabilized rate of decline was observed. Measured infiltration rates vary from 194.14 to 1,270.00 mm/hr (7.64 in/hr – 50.00 in/hr) (Table 4-2).

\textsuperscript{29} Geoprobe Systems Macro-Core®
Figure 4-3. Bioinfiltration basin monitoring sites in Aiken, SC, USA with (a) turf cover near intersection of Park, Union, and Fairfield streets and (b) vegetated cover near intersection of Chesterfield, Richland, and Park streets.
Table 4-2. Selected geotechnical data for native soil samples.

<table>
<thead>
<tr>
<th>Location (SLSCODE)</th>
<th>Sample depth (m)</th>
<th>Infiltration rate (mm hr(^{-1}))</th>
<th>Soil classification</th>
<th>Atterberg limits</th>
<th>Natural moisture (%)</th>
<th>Percent passing No. 200 sieve</th>
<th>Gravel (%)</th>
<th>Sand (%)</th>
<th>Silt (%)</th>
<th>Clay (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-1 (FuB)</td>
<td>0.00 – 1.82</td>
<td>426.72</td>
<td>silty fine to medium sand</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.61 – 1.23</td>
<td>-</td>
<td>-</td>
<td>NP</td>
<td>NP</td>
<td>7.2</td>
<td>16.3</td>
<td>0.0</td>
<td>83.7</td>
<td>16.3</td>
</tr>
<tr>
<td>B-2 (FuB)</td>
<td>0.00 – 1.82</td>
<td>365.76</td>
<td>silty fine to medium sand</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>B-3 (FuB)</td>
<td>0.00 – 1.82</td>
<td>1,270.00</td>
<td>silty clayey fine to medium sand</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>B-4 (FuB)</td>
<td>0.00 – 1.82</td>
<td>579.12</td>
<td>silty clayey fine to medium sand</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>B-5 (Fub)</td>
<td>0.00 – 1.82</td>
<td>548.64</td>
<td>silty fine to medium sand</td>
<td>0.61</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.61 – 1.23</td>
<td>-</td>
<td>-</td>
<td>NP</td>
<td>NP</td>
<td>5.2</td>
<td>21.7</td>
<td>0.0</td>
<td>78.3</td>
<td>21.7</td>
</tr>
<tr>
<td>B-6 (OrA)</td>
<td>0.00 – 1.82</td>
<td>104.14</td>
<td>silty fine to medium sand</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>1.23 – 1.83</td>
<td>-</td>
<td>-</td>
<td>35</td>
<td>26</td>
<td>9</td>
<td>43.6</td>
<td>0.2</td>
<td>56.2</td>
<td>43.6</td>
</tr>
<tr>
<td>B-7 (OrA)</td>
<td>0.00 – 1.82</td>
<td>274.32</td>
<td>silty clayey fine to medium sand</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>B-8 (OrA)</td>
<td>0.00 – 1.82</td>
<td>426.72</td>
<td>clayey fine to medium sand</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>0.30 – 1.52</td>
<td>-</td>
<td>-</td>
<td>33</td>
<td>22</td>
<td>11</td>
<td>20.2</td>
<td>0.0</td>
<td>53.0</td>
<td>47.0</td>
</tr>
<tr>
<td>B-9 (OrA)</td>
<td>0.00 – 1.82</td>
<td>309.88</td>
<td>silty clayey fine to medium sand</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>B-10 (FuB)</td>
<td>0.00 – 1.82</td>
<td>213.36</td>
<td>clayey fine to medium sand</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Native soils in vegetative medians. Sample Date: 6/8/09

<table>
<thead>
<tr>
<th>Location (SLSCODE)</th>
<th>Sample depth (m)</th>
<th>Infiltration rate (mm hr(^{-1}))</th>
<th>Soil classification</th>
<th>Atterberg limits</th>
<th>Natural moisture (%)</th>
<th>Percent passing No. 200 sieve</th>
<th>Gravel (%)</th>
<th>Sand (%)</th>
<th>Silt (%)</th>
<th>Clay (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-1 (FuB)</td>
<td>0.30 – 0.91</td>
<td>-</td>
<td>silty fine to medium sand</td>
<td>NP</td>
<td>NP</td>
<td>6.0</td>
<td>13.6</td>
<td>5.7</td>
<td>80.7</td>
<td>13.6</td>
</tr>
<tr>
<td>C-2 (FuB)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C-3 (OrA)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>NP</td>
<td>NP</td>
<td>9.2</td>
<td>20.4</td>
<td>2.2</td>
<td>77.4</td>
<td>20.4</td>
</tr>
<tr>
<td>C-4 (OrA)</td>
<td>0.30 – 1.52</td>
<td>-</td>
<td>silty fine to medium sand</td>
<td>NP</td>
<td>NP</td>
<td>9.2</td>
<td>20.4</td>
<td>2.2</td>
<td>77.4</td>
<td>20.4</td>
</tr>
<tr>
<td>C-5 (OrA)</td>
<td>0.91 – 1.52</td>
<td>-</td>
<td>silty clayey fine to medium sand</td>
<td>23</td>
<td>16</td>
<td>7</td>
<td>39.0</td>
<td>1.5</td>
<td>59.5</td>
<td>39.0</td>
</tr>
<tr>
<td>C-6 (OrA)</td>
<td>0.30 – 0.91</td>
<td>-</td>
<td>silty fine to medium sand</td>
<td>NP</td>
<td>NP</td>
<td>1.4</td>
<td>22.9</td>
<td>0.1</td>
<td>77.0</td>
<td>22.9</td>
</tr>
<tr>
<td>C-7 (OrA)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>C-8 (OrA)</td>
<td>0.30 – 1.52</td>
<td>-</td>
<td>silty clayey fine to medium sand</td>
<td>20</td>
<td>14</td>
<td>7</td>
<td>30.3</td>
<td>1.7</td>
<td>68.0</td>
<td>30.3</td>
</tr>
<tr>
<td>C-9 (FuB)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>20</td>
<td>15</td>
<td>5</td>
<td>40.8</td>
<td>0.0</td>
<td>59.2</td>
<td>40.8</td>
</tr>
<tr>
<td>C-10 (OrA)</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Native soils under asphalt. Sample Date: 6/9/09

Note: LL = Liquid Limit, PL = Plastic Limit, PI = Plasticity Index, NP = Non-Plastic
Engineered Media Sampling and Composition Analysis

Samples of the engineered bioinfiltration soil media (BSM) were obtained after the media had been graded into the excavated cell. Laboratory tests include soil gradation, chemical composition, soil structure, organic matter, and density. Soil gradation tests were performed according to ASTM D1140 (ASTM 2006; Table 4-3). Results from soil gradation analysis were tabulated using Gradistat (Blott and Pye 2001; Table 4-3). Media mixtures varied in volumetric proportions of gravel, sand, soil, and compost. Within these mixtures, the gravel component ranged from 0.0% to 1.3%, sand from 81.2% to 87.9%, clay from 1.2% to 4.6%, and silt from 6.6% to 17.5% (Table 4-3).

Engineered Media Chemical Analysis

Chemical analysis of engineered media was conducted by the Clemson University Agricultural Service Laboratory using its standards approved analytical procedures and documented Quality Assurance/Quality Control procedures. Standard soil tests were conducted by the laboratory to determine soil and buffer pH; acidity; total extractable phosphorus (P), potassium (K), calcium (Ca), magnesium (Mg), zinc (Zn), manganese (Mn), copper (Cu), boron (B), and sodium (Na); and cation exchange capacity (CEC). Upon receipt, test laboratory personnel logged in the samples and assigned each one a unique 7-digit identification number. The samples were placed on drying racks with a fan blowing room temperature air across them to facilitate complete drying. After drying, the soil samples were screened through a 10-mesh (2-mm) screen, ground to reduce the particle size, and mixed uniformly before analysis. Chemical properties are summarized in Table 4-4.
Table 4-3. Selected physical properties for engineered media samples.

<table>
<thead>
<tr>
<th>Location (SLSCODE)</th>
<th>Sampling date</th>
<th>Sample depth (m)</th>
<th>Saturated Hydraulic Conductivity* (mm hr⁻¹)</th>
<th>Organic matter (%)</th>
<th>Soil classification</th>
<th>Percent passing No. 200 sieve</th>
<th>Gravel (%)</th>
<th>Sand (%)</th>
<th>Silt (%)</th>
<th>Clay (%)</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-1 (OrA)</td>
<td>09/09/10</td>
<td>0.05 – 0.15</td>
<td>132.68</td>
<td>1.91</td>
<td>loamy sand</td>
<td>-</td>
<td>83.7</td>
<td>14.9</td>
<td>1.4</td>
<td></td>
<td>Gradistat**</td>
</tr>
<tr>
<td></td>
<td>09/03/10</td>
<td>0.05 – 0.15</td>
<td>109.49</td>
<td>1.50</td>
<td></td>
<td>14.8</td>
<td>0.0</td>
<td>85.2</td>
<td>11.5</td>
<td>3.3</td>
<td>ASTM D1140***</td>
</tr>
<tr>
<td>M-2 (OrA)</td>
<td>09/09/10</td>
<td>0.05 – 0.15</td>
<td>136.78</td>
<td>1.94</td>
<td>loamy sand</td>
<td>-</td>
<td>85.7</td>
<td>13.0</td>
<td>1.3</td>
<td></td>
<td>Gradistat</td>
</tr>
<tr>
<td></td>
<td>09/03/10</td>
<td>0.05 – 0.15</td>
<td>125.57</td>
<td>1.60</td>
<td></td>
<td>13.2</td>
<td>0.3</td>
<td>86.5</td>
<td>10.2</td>
<td>3.0</td>
<td>ASTM D1140</td>
</tr>
<tr>
<td>M-3 (OrA)</td>
<td>09/09/10</td>
<td>0.05 – 0.15</td>
<td>126.56</td>
<td>1.98</td>
<td>sand</td>
<td>-</td>
<td>81.2</td>
<td>17.5</td>
<td>1.4</td>
<td></td>
<td>Gradistat</td>
</tr>
<tr>
<td></td>
<td>09/03/10</td>
<td>0.05 – 0.15</td>
<td>115.91</td>
<td>1.80</td>
<td></td>
<td>10.8</td>
<td>1.3</td>
<td>87.9</td>
<td>6.6</td>
<td>4.2</td>
<td>ASTM D1140</td>
</tr>
<tr>
<td>M-4 (OrA)</td>
<td>09/09/10</td>
<td>0.05 – 0.15</td>
<td>110.81</td>
<td>1.84</td>
<td>loamy sand</td>
<td>-</td>
<td>85.2</td>
<td>13.6</td>
<td>1.2</td>
<td></td>
<td>Gradistat</td>
</tr>
<tr>
<td></td>
<td>09/03/10</td>
<td>0.05 – 0.15</td>
<td>99.57</td>
<td>1.50</td>
<td></td>
<td>14.9</td>
<td>0.8</td>
<td>84.3</td>
<td>10.3</td>
<td>4.6</td>
<td>ASTM D1140</td>
</tr>
<tr>
<td>M-5 (OrA)</td>
<td>08/19/10</td>
<td>0.05 – 0.15</td>
<td>126.56</td>
<td>-</td>
<td></td>
<td>-</td>
<td>81.2</td>
<td>17.1</td>
<td>1.8</td>
<td></td>
<td>Gradistat</td>
</tr>
<tr>
<td>M-6 (OrA)</td>
<td>08/19/10</td>
<td>0.05 – 0.15</td>
<td>126.56</td>
<td>-</td>
<td>loamy sand</td>
<td>-</td>
<td>81.3</td>
<td>17.2</td>
<td>1.6</td>
<td></td>
<td>Gradistat</td>
</tr>
<tr>
<td></td>
<td>09/03/10</td>
<td>0.05 – 0.15</td>
<td>110.81</td>
<td>2.00</td>
<td></td>
<td>14.8</td>
<td>0.0</td>
<td>85.2</td>
<td>10.9</td>
<td>3.9</td>
<td>ASTM D1140</td>
</tr>
<tr>
<td>M-7 (OrA)</td>
<td>08/19/10</td>
<td>0.05 – 0.15</td>
<td>109.79</td>
<td>1.18</td>
<td>-</td>
<td>-</td>
<td>85.3</td>
<td>13.4</td>
<td>1.4</td>
<td></td>
<td>Gradistat</td>
</tr>
</tbody>
</table>

*Estimation based on procedure described by Saxton and Rawls, 2006.
**Method described by Blott and Pye, 2001.
***Method described by ASTM, 2006.
### Table 4-4. Selected chemical properties for engineered media samples.

<table>
<thead>
<tr>
<th>Location</th>
<th>Sampling date</th>
<th>Sample depth (m)</th>
<th>pH</th>
<th>Acidity (meq 100g⁻¹)</th>
<th>CEC (meq 100g⁻¹)</th>
<th>P (kg ha⁻¹)</th>
<th>K</th>
<th>Na</th>
<th>Ca</th>
<th>Mg</th>
<th>Zn</th>
<th>Mn</th>
<th>Cu</th>
<th>B</th>
<th>S</th>
<th>Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>M-1</td>
<td>09/09/10</td>
<td>0.05 – 0.15</td>
<td>6.4</td>
<td>1.2</td>
<td>3.6</td>
<td>21.3</td>
<td>49.3</td>
<td>16.8</td>
<td>906.8</td>
<td>88.5</td>
<td>7.4</td>
<td>14.6</td>
<td>2.2</td>
<td>0.6</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>09/13/10</td>
<td>0.05 – 0.15</td>
<td>6.2</td>
<td>0.8</td>
<td>4.3</td>
<td>48.2</td>
<td>69.5</td>
<td>-</td>
<td>1018.9</td>
<td>94.2</td>
<td>12.3</td>
<td>26.9</td>
<td>3.5</td>
<td>0.6</td>
<td>?</td>
<td>31.4</td>
</tr>
<tr>
<td></td>
<td>10/11/10</td>
<td>0.05 – 0.15</td>
<td>6.2</td>
<td>-</td>
<td>4.0</td>
<td>13.5</td>
<td>47.1</td>
<td>-</td>
<td>924.7</td>
<td>77.3</td>
<td>6.3</td>
<td>9.0</td>
<td>1.7</td>
<td>0.4</td>
<td>?</td>
<td>45.9</td>
</tr>
<tr>
<td></td>
<td>09/03/10</td>
<td>0.05 – 0.15</td>
<td>6.1</td>
<td>-</td>
<td>3.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td>M-2</td>
<td>09/09/10</td>
<td>0.05 – 0.15</td>
<td>6.2</td>
<td>1.2</td>
<td>3.8</td>
<td>22.4</td>
<td>52.7</td>
<td>17.9</td>
<td>981.9</td>
<td>96.4</td>
<td>7.6</td>
<td>16.8</td>
<td>2.4</td>
<td>0.7</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>09/13/10</td>
<td>0.05 – 0.15</td>
<td>-</td>
<td>0.6</td>
<td>3.8</td>
<td>58.3</td>
<td>66.1</td>
<td>-</td>
<td>983.0</td>
<td>93.0</td>
<td>11.7</td>
<td>26.9</td>
<td>3.3</td>
<td>0.4</td>
<td>?</td>
<td>47.1</td>
</tr>
<tr>
<td></td>
<td>10/11/10</td>
<td>0.05 – 0.15</td>
<td>-</td>
<td>-</td>
<td>3.6</td>
<td>15.7</td>
<td>54.9</td>
<td>-</td>
<td>925.8</td>
<td>81.8</td>
<td>5.9</td>
<td>12.3</td>
<td>1.8</td>
<td>0.7</td>
<td>?</td>
<td>41.5</td>
</tr>
<tr>
<td></td>
<td>09/03/10</td>
<td>0.05 – 0.15</td>
<td>6.1</td>
<td>-</td>
<td>3.6</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td>M-3</td>
<td>09/09/10</td>
<td>0.05 – 0.15</td>
<td>7.1</td>
<td>0.4</td>
<td>3.6</td>
<td>17.9</td>
<td>51.6</td>
<td>14.6</td>
<td>1212.8</td>
<td>106.5</td>
<td>7.7</td>
<td>20.2</td>
<td>1.9</td>
<td>0.4</td>
<td>?</td>
<td>20.2</td>
</tr>
<tr>
<td></td>
<td>09/13/10</td>
<td>0.05 – 0.15</td>
<td>-</td>
<td>0.6</td>
<td>5.6</td>
<td>40.4</td>
<td>78.5</td>
<td>-</td>
<td>1694.7</td>
<td>141.2</td>
<td>14.2</td>
<td>42.6</td>
<td>3.7</td>
<td>0.3</td>
<td>?</td>
<td>40.4</td>
</tr>
<tr>
<td></td>
<td>10/11/10</td>
<td>0.05 – 0.15</td>
<td>-</td>
<td>-</td>
<td>4.7</td>
<td>16.8</td>
<td>58.3</td>
<td>-</td>
<td>1383.1</td>
<td>103.1</td>
<td>8.1</td>
<td>15.7</td>
<td>1.7</td>
<td>0.6</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>09/03/10</td>
<td>0.05 – 0.15</td>
<td>7.1</td>
<td>-</td>
<td>4.3</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td>M-4</td>
<td>09/09/10</td>
<td>0.05 – 0.15</td>
<td>6.2</td>
<td>1.2</td>
<td>3.6</td>
<td>20.2</td>
<td>51.6</td>
<td>19.1</td>
<td>895.6</td>
<td>90.8</td>
<td>7.2</td>
<td>12.3</td>
<td>2.0</td>
<td>0.6</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>09/13/10</td>
<td>0.05 – 0.15</td>
<td>-</td>
<td>0.8</td>
<td>4.4</td>
<td>50.4</td>
<td>68.4</td>
<td>-</td>
<td>1037.9</td>
<td>97.5</td>
<td>11.3</td>
<td>23.5</td>
<td>3.4</td>
<td>0.7</td>
<td>?</td>
<td>56.0</td>
</tr>
<tr>
<td></td>
<td>10/11/10</td>
<td>0.05 – 0.15</td>
<td>-</td>
<td>-</td>
<td>4.3</td>
<td>14.6</td>
<td>48.2</td>
<td>-</td>
<td>1036.8</td>
<td>94.2</td>
<td>5.6</td>
<td>7.8</td>
<td>1.8</td>
<td>0.6</td>
<td>?</td>
<td>40.4</td>
</tr>
<tr>
<td></td>
<td>09/03/10</td>
<td>0.05 – 0.15</td>
<td>6.0</td>
<td>-</td>
<td>3.9</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td>M-5</td>
<td>08/19/10</td>
<td>0.05 – 0.15</td>
<td>6.1</td>
<td>1.2</td>
<td>3.8</td>
<td>24.7</td>
<td>49.3</td>
<td>11.2</td>
<td>918.0</td>
<td>93.0</td>
<td>7.4</td>
<td>13.4</td>
<td>2.1</td>
<td>0.6</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>09/16/10</td>
<td>0.05 – 0.15</td>
<td>-</td>
<td>-</td>
<td>3.8</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td>M-6</td>
<td>08/19/10</td>
<td>0.05 – 0.15</td>
<td>6.0</td>
<td>1.2</td>
<td>3.8</td>
<td>21.3</td>
<td>50.4</td>
<td>11.2</td>
<td>961.69</td>
<td>94.2</td>
<td>8.4</td>
<td>14.6</td>
<td>1.9</td>
<td>0.7</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>09/16/10</td>
<td>0.05 – 0.15</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>09/03/10</td>
<td>0.05 – 0.15</td>
<td>6.1</td>
<td>-</td>
<td>3.4</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>?</td>
<td>-</td>
</tr>
<tr>
<td>M-7</td>
<td>08/19/10</td>
<td>0.05 – 0.15</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>?</td>
<td>-</td>
</tr>
</tbody>
</table>
The pH of all soil samples was determined by equilibrating 20 g of each soil with 20 ml of deionized water for a minimum of 1 h and then measuring the pH with a calibrated AS-3000 Dual pH Analyzer. Buffer pH was determined for these same samples using the Adams-Evans buffer method (Moore and Franklin 2002) and the pH analyzer. Soil acidity (meq/100 g) was calculated by the test laboratory as 8 times the difference between pH 8 and the measured buffer pH, which accounts for the soil mass used in the buffer pH test. Mineral analyses (P, K, Ca, Mg, Na, Zn, Mn, Cu, B) were determined using a Mehlich No. 1 extraction solution and element quantification by inductively coupled plasma optical emission spectroscopy (ICP-OES) (Isaac and Donohue, 1983; Jones, 2001). Following the test laboratory’s standard procedure for soils in South Carolina, CEC was estimated from the sum of acidity plus all base cation (K, Ca, Mg, Na) concentrations in the Mehlich 1 extract expressed in meq/100 g. Note that this laboratory reported CEC value is an estimate of the actual CEC because it is calculated from the Mehlich 1 extractable cations and the calculated soil acidity (Mullins and Heckendorn 2009). For example, for high pH soils or soils with high levels of soluble salts, the CEC estimated by this procedure can be erroneously high (Mullins and Heckendorn 2009). However, for most acidic soils in the Southeast U.S., the value estimated by this procedure can be considered an effective CEC since it is the CEC at the current soil pH (Mullins and Heckendorn 2009). The test laboratory calculated total base saturation as the percent of estimated CEC occupied by all base cations measured in the Mehlich 1 extract.
Rainfall, Discharge and Soil Moisture Content Monitoring Procedure

Rainfall data was collected with a Campbell Scientific TE525WS tipping bucket rain gauge capable of detecting rainfall at 0.254 mm (0.01 in.) increments. Rainfall measurements were collected in region unobstructed by trees or buildings. Monitoring locations are shown in Figure 4-1. Data collection occurred on minute intervals. Data collection began in June of 2010 and is ongoing at the time of writing. Data for this study is limited to the period of June 2011 – May 2012.

Water level measurements were taken in selected BRC sites (Figure 4-1). All sites were constructed with concrete storm-sewer outlet structures to provide an overflow outlet, limiting maximum ponding depth. These structures receive water when a certain ponding depth is achieved or, in certain cases, when basin under drains are enabled. Water level measurements were recorded inside these concrete structures using a YSI 600LS multi-parameter sonde with depth, temperature, and conductivity measurements. All BRC sites with pipe connections to upstream storm-sewer systems incorporate a concrete cistern structure to provide controlled inflow of storm-water. These cisterns were monitored at selected sites using YSI 600LS multi-parameter sondes. Measurements were recorded on 5-minute intervals.

Soil moisture measurements were taken in selected BRC sites (Figure 4-1). The measurement assembly consists of a vertical profile of Decagon 5TE and Decagon 5TM volumetric water content sensors. These sensors measure $VWC_{sat}$ at a resolution of 0.0008 m$^3$m$^{-3}$ below 0.50 m$^3$m$^{-3}$ and 0.009 m$^3$m$^{-3}$ above 0.65 m$^3$m$^{-3}$ (Personal Communication Douglas Cobos, Decagon Devices). Sensor spacing and orientation was
controlled by mounting the base of the sensor in a 25.4 mm (1 in) PVC pipe prior to installation (Figure 4-4b). Individual sensors were positioned to orient vertically once installed in the soil. Sensors were placed at depth intervals of 15.24 - 30.48 cm (6 - 12 in) to monitor multiple depths of engineered soil media and the basin subsurface (native soil). Monitoring locations and depths are described in Table 4-5. The number of sensors in the profile depended on the depth of the basin at the installation point. Sensor spacing is necessary to prevent current from one probe from being detected by a second probe. To minimize media disturbance during installation, 25.4 cm (10 in) PVC sleeves are used as a placeholder for the sensor assembly prior to media infill. Once media installation was complete, the soil moisture sensing assembly was lowered into the sleeve to a predetermined depth. The sleeve was removed and the native soil and soil media was carefully replaced. Additionally, four soil moisture sensors were placed within BRCs, but outside of the ponding area. These sensors were located near mature trees at shallow depths to monitor root zone moisture (Table 4-5).

Instruments to measure rainfall, soil moisture content, and water level are controlled by a purpose-built embedded device called a Motestack (Figure 4-4a). A Motestack is a participant in a wireless sensor network, communicating via the IEEE 802.15.4 wireless communications standard. A Digi ConnectPort X2 gateway device is used to bridge 802.15.4 traffic onto an 802.11 (WiFi) network. The WiFi network uses a mesh configuration of Anaptyx access points to support multi-path routing for

30 http://standards.ieee.org/
32 http://anaptyx.com/
Table 4-5. Soil moisture monitoring assembly configuration by site and depth.

<table>
<thead>
<tr>
<th>Bioinfiltration Basin Code*</th>
<th>Site Number</th>
<th>Surface Elevation (m)</th>
<th>Soil Moisture Probe Depth (cm)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Probe 1</td>
</tr>
<tr>
<td>PNL</td>
<td>1</td>
<td>156.12</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>156.19</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>156.06</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>155.98</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>156.05</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>6**</td>
<td></td>
<td>20.32</td>
</tr>
<tr>
<td></td>
<td>7**</td>
<td></td>
<td>20.32</td>
</tr>
<tr>
<td>CRPS</td>
<td>1</td>
<td>156.08</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>156.52</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>156.39</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>156.42</td>
<td>15.24</td>
</tr>
<tr>
<td>CRPN</td>
<td>5</td>
<td>157.25</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>157.35</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>157.44</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>157.28</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>156.94</td>
<td>15.24</td>
</tr>
<tr>
<td>PCN</td>
<td>1</td>
<td>155.11</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>155.08</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>155.65</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>155.60</td>
<td>15.24</td>
</tr>
<tr>
<td>PUF</td>
<td>1</td>
<td>153.02</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>153.00</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>153.11</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>152.99</td>
<td>15.24</td>
</tr>
<tr>
<td></td>
<td>5**</td>
<td></td>
<td>20.32</td>
</tr>
<tr>
<td></td>
<td>6**</td>
<td></td>
<td>20.32</td>
</tr>
</tbody>
</table>

* Abbreviation for street intersection. See Figure 1.
** Root zone monitoring site.
Figure 4-4. Soil water content monitoring devices including the a) Motestack data acquisition device and a b) soil moisture monitoring assembly prior to infill of engineered bioinfiltration media (Image: Christopher Bellamy, Clemson University, 2009).
increased fault tolerance. The WiFi network is linked to an Internet connection provided by the City of Aiken. Data is streamed to the Clemson University campus where it is integrated into the Intelligent River monitoring cyberinfrastructure (Eidson et al. 2010b).

**Preliminary Observation Data Processing and Analysis**

Prior to analysis, all soil moisture data underwent a calibration adjustment to transform dielectric permittivity ($\varepsilon$) to Volumetric Water Content ($VWC, \ m^3 \ m^{-3}$). Decagon Devices, Inc. generated Equation (1) based on a laboratory analysis of media samples.

$$\theta = -0.000263\varepsilon^2 + 0.0257\varepsilon - 0.0695 \quad (4-1)$$

Where $\theta$ is the volumetric water content with units ($m^3 \ m^{-3}$), and $\varepsilon$ is the measured dielectric permittivity returned by the probe. Basic quality control (QC) was performed on observation data using the MATLAB\textsuperscript{33} analysis software. Rainfall data collected by the Campbell Scientific data logger required no removal of outliers or invalid data. Soil moisture and flow data required considerable preliminary processing before it could be incorporated into later analysis steps. This was due to the volume of data, the frequency of missing data, and the presence of erroneous values. The first QC step involved a check to ensure all measurements fell within the manufacturers specified measurement range. Data outside of this range was discarded. Several techniques were evaluated for outlier detection, e.g., box (and whisker) plots. However, the highly positively skewed distributions made automated outlier detection difficult, even with data transformations, e.g., log transform. Methods involving time-series analysis, e.g., exponentially weighted

moving averages, likewise yielded poor results. The data exhibited long periods of consistent values followed by abrupt fluctuations, a scenario that is typical of soil moisture data. Missing data points further complicated Quality Control procedures. Missing data resulted in sequences of observations appearing to have abrupt variations (spikes) when the true measurement values would have shown a more gradual fluctuation. Rather than risk removing valid data points; no outlier removal was performed on either the soil moisture or water depth measurements.

Soil moisture and water depth measurements were resampled to align on regular intervals to simplify later analysis. This step facilitated comparison among measurements and introduced a minimal amount of smoothing over the data. Data was aligned to five-minute intervals. In the case of soil moisture and water depth data, multiple measurements within the interval window were averaged, ignoring missing values. If no measurements were available for a given window, a value of Not-a-Number (NaN) designation was assigned. Rainfall data were aligned to the same five-minute intervals based on summing, rather than averaging.

Additional exploratory data analysis was performed to validate sensor function, identify notable rainfall events and ascertain the extent of missing data present during each rainfall event. This includes univariate statistics, histograms, and Quantile-Quantile (Q-Q) plots to evaluate untransformed and transformed data against a normal distribution.

**Spatial Structure of Infiltration Properties Analysis**

Infiltration rates of soils exhibit high spatial variability (Beven 2001; Greminger et al. 1985), particularly in disturbed urban soils (Pitt et al. 2000). Infiltration rate in soil
is dependent, among other things, on soil water content. For consistency, this analysis uses saturated hydraulic conductivity ($K_{sat}$) to describe infiltration rate at saturation. Spatial structure analysis of $K_{sat}$ is evaluated using three methods. Univariate exploratory data analysis (EDA) is used to evaluate measures of central tendency and facilitate qualitative assessment of the distribution of measured values. A spatial autocorrelation statistical test is applied to evaluate whether observed $K_{sat}$ in our AOI is spatially autocorrelated. Finally, a geostatistical approach using a semivariogram is used to evaluate the influence of scale on the spatial autocorrelation. All analysis was performed using the MATLAB data analysis and programming environment.

The spatial autocorrelation statistical test is based on a null hypothesis of Complete Spatial Randomness (CSR). Spatially structured processes, e.g., geologic, climate, determine the characteristics of soils at large-scales. This implies that spatial autocorrelation is always present in soil properties. At smaller scales, non-spatial processes may be the predominate source of variation, leading to Type II statistical errors when applying spatial autocorrelation tests. Previous studies of spatial autocorrelation of $K_{sat}$ have demonstrated this dependence on scale. Processes like topography and soil series dominate at large scales and biological processes dominate at local scales (Sobieraj et al. 2004). At the spatial scale of the AOI, biological processes or urban disturbance are expected to be more significant drivers of variation than topography or soil unit. To evaluate whether spatial structure is present in the AOI, the Moran’s $I$ statistical test for spatial autocorrelation is applied (Moran P.A.P. 1950). Moran’s $I$ is calculated as follows:
Where \( n \) is the number of observations on variable \( x \) at locations \( i, j \). The mean of \( x \) is shown by \( \bar{x} \). The \( w_{ij} \) term is the weight matrix, with \( S_0 \) being the sum of all elements in weight matrix. The weighting is based on the inverse distance of features. Moran’s I will vary from negative one to positive one. If no spatial autocorrelation occurs, than Moran’s I will take on the expected value shown in Equation 4-4, which approaches zero as sample size increases. A positive Moran’s I indicates positive spatial autocorrelation, while a negative value indicates negative spatial autocorrelation.

The degree of spatial autocorrelation is dependent on scale (Goodchild 1986). Sobieraj et al. (2004) found a lack of spatial structure in observed \( K_{sat} \) based on semivariogram analysis at scales of 0.25, 1, 10, and 25 meters. We evaluate the observed \( K_{sat} \) at similar scales for our AOI using a semivariogram. The semivariogram estimator \( \gamma(h) \) is described by Goovaerts (1997) as follows:

\[
\gamma(h) = \frac{1}{2N(h)} \sum_{i=1}^{N(h)} [z(x_i) - z(x_i + h)]^2
\]  

Where \( \gamma(h) \) measures the average dissimilarity between data separated by vector \( h \). Vector \( h \) is the lag distance between observed value at location \( z(x_i) \). The number of
pairs separated by a given lag distance is $N(h)$. Sample site distances are less than 1 km. A lag distance of 10 meters is chosen to avoid summarization (binning) of distance pairs.

**Trend Detection in Effective Volumetric Water Content at Saturation**

Bioinfiltration systems are a relatively recent storm water management technology with limited long-term monitoring data to evaluate how bioinfiltration systems change over time (Emerson and Traver 2009). Previous studies have identified compaction and sedimentation as two of the leading causes of degradations in performance (Lindsey et al. 1992). Soil compaction from heavy machinery traffic and excavation techniques during installation can reduce infiltration capacity (Brown and Hunt 2010). Further compaction may occur after installation due to landscaping and foot traffic, particularly if the bioinfiltration basin is covered with grass turf (Davis et al. 2009). Thompson et al. (2008) found that soil wetting lead to compaction, increased bulk density, and decreased moisture-holding capacity; and was dependent on the engineered media composition.

Emerson and Traver (2009) describe a *bioinfiltration performance indicator* suitable for long-term monitoring of bioinfiltration basins. This indicator is based on the receding limb of a ponding depth measurement. The measurement is taken during and following a rain event. An estimate of $K_{sat}$ is made based on the slope of the recession limb versus time, subject to a correction for media porosity. Emerson and Traver (2009) use this approach to identify a strong seasonal signal in infiltration performance due to temperature-induced viscosity changes in water, i.e. summer increasing $K_{sat}$, winter decreasing $K_{sat}$. The performance indicator does not account for matric-suction early-
time infiltration or the effects of air entrapment (Emerson and Traver 2009). Additionally, no accommodation is made for sustained rainfall events, where continuing rainfall input may decrease the ponding depth recession rate. Thus, sustained rainfall scenarios may influence the receding limb approach. This may partially account for the seasonality findings. Shorter duration, higher intensity rainfalls occur more frequently in the summer, whereas longer sustained rainfall are more likely in the winter. The use of ponding depth limits this approach bioinfiltration basins that experience measureable ponding. Sites with high infiltration rates or under drain systems may not generate enough data to produce performance measurements.

An alternative method of monitoring performance is presented based on the relationship between the saturation limit of volumetric water content ($VWC_{sat}$) and the effective soil porosity of the media. A declining trend in effective soil porosity corresponds with degradation of bioinfiltration performance. Trends in effective porosity can be evaluated based on $VWC_{sat}$ measurements over time. The relationship is between porosity and observed $VWC_{sat}$ can be obscured by dynamic factors, e.g., hysteresis effects. The presence of air bubbles trapped in the media or the presence of clay, which may swell upon wetting (ASCE 1996). Rainfall intensity and magnitude influence observed $VWC_{sat}$ as faster ponding or greater ponding depths affect air entrapment within the media. Another limitation of the $VWC_{sat}$ approach is the assumption that saturation will occur under heavy or prolonged rainfall. In media with high hydraulic conductivity, the saturation point may not necessarily be reached.
Saturated volumetric water content was measured using the Decagon 5TE/TM sensors. Identification of saturating conditions was performed by first evaluating a twenty-four hour window of VWC measurements following a rainfall event. Rainfall events were identified based on rainfall thresholds. Low thresholds were used to avoid ruling out saturation conditions caused by low magnitude rainfalls occurring during periods of high antecedent moisture. Rainfall events within 3 days of another event were considered as a single event, with the greater of the rainfall magnitudes being chosen as the representative value. A local maximum was identified during the rainfall event period. Local maximums were evaluated visually and by comparison with water depth measurements obtained within the basin outflow structure. This analysis step resulted in an identification of saturating rainfall events and corresponding $VWC_{sat}$ maximums for every soil moisture sensor.

Statistical analysis was performed to identify whether $VWC_{sat}$ measurements exhibited a detectable trend over time. Exploratory data analysis was performed to evaluate the distribution of measured values and the validity of normality assumptions. A linear model was fit to the measured $VWC_{sat}$ response as a function of time. Homogeneity was validated based on visual inspection of residuals versus fitted value plots. The slope of the trend line was evaluated using a t-statistic. The linear model provides an easily interpretable result, but is subject to limitations. The linear model is not able to isolate the influence of depth, site, or native soil vs. engineered media factors on the VWC response. Furthermore, the use of repeated measurements on the same experimental units (soil region surrounding sensor), introduces a violation of the
independence of observation assumption. An alternate approach based on a linear mixed effects model was applied to evaluate the influence of fixed effects and the importance of the independence assumption. Time, depth, and native soil versus engineered media are treated as fixed effects while site is treated as a random effect. The model was fit using REML. Unlike repeated measures ANOVA approaches, mixed effects models support missing and unbalanced data. Combinations of effects and interactions were evaluated to find the optimal model configuration. Normality and homogeneity were checked by visual inspection of Q-Q and residual plots. All analysis was performed using the R statistical analysis software (R Core Development Team 2012) and the R package ‘nlme’ (Pinheiro et al. 2012).

**Runoff and Infiltration Production and Routing Model**

Two methods are used to model infiltration and runoff excess for the study area. The first approach employs the Green-Ampt (GA) equation to calculate infiltration and runoff production for each raster cell of the bioinfiltration basin and the surrounding pervious area (BRC). The GA component operates within a GIS framework and supports spatially varying soil/media properties, unsteady rainfall inputs, and runoff excess flow routing. The flow routing component implements the D-infinity multiple flow direction algorithm (Tarboton 1997). A second infiltration modeling approach is use to calculate rainfall excess originating from regions of impervious cover directly connected to the bioinfiltration basin. This method uses the empirically derived Curve Number (CN) methodology to generate a unit hydrograph for each connected impervious area (US SCS
A ponding component is incorporated into the model to allow the redistribution of rainfall excess volume across a basin depression at each model time step.

First described in 1911, the GA equation is widely used to model one-dimensional vertical movement of water into unsaturated soils (Green and Ampt 1911; Browne et al. 2008; Dussaillant et al. 2004; Heasom et al. 2006). It offers a simplified solution to Richard’s equation. Richard’s equation describes water movement through unsaturated soils. However, it does not have an analytical solution under most circumstances. The GA simplification is based on assumptions about the physical processes of infiltration. Notably, the GA equation assumes a sharply defined wetting front that divides the unsaturated and saturated zones of a column of soil. The unsaturated zone is defined by constant initial volumetric water content, while the saturated zone is assumed to have volumetric water content equal to its effective porosity. The GA infiltration rate and infiltration depth is calculated by:

\[
f_t = \begin{cases} 
  \frac{R}{K_s \left( \psi \left( VWC_{sat} - VWC_{init} \right) \right)} + 1, & t \leq t_p \\
  \left( VWC_{sat} - VWC_{init} \right), & t > t_p 
\end{cases}
\]

(4-6)

\[
F_{t+\Delta t} = F_t + \min \left( R, f_t \right)
\]

(4-7)

Where:

\( f_t = \text{infiltration rate} \ (L \ T^{-1}) \)

\( R_t = \text{rainfall} \ (L) \)

\( K_s = \text{hydraulic conductivity at saturation} \)

\( \psi = \text{wetting front suction} \ (L) \)

\( VWC_{sat} = \text{effective soil porosity} \ (L \ L^{-1}) \)
\[ V W C_{\text{initial}} = \text{initial soil water content} \ \ (L \ L^{-1}) \]

\[ F_t = \text{infiltration depth} \ \ (L) \]

\[ t_p = \text{time to surface ponding} \]

Parameters are based on soil properties. These properties may be directly observed or obtained through estimation techniques (e.g., pedotransfer functions [Rawls and Brakensiek 1982]). Skaggs and Khalel (1982) performed a sensitivity analysis of the GA equation and found infiltration rate to be most sensitive to the porosity and hydraulic conductivity. Parameter estimates for this model are based on measured soil properties and estimation techniques described in the literature.

Data preparation steps are performed using ESRI’s ArcGIS® Desktop\(^{34}\) software. This step included estimation of soil property surfaces and delineation of BRC and directly connected impervious areas. The GA and CN infiltration, rainfall excess, and runoff routing algorithms were developed using Python, the ESRI ArcPy library, and the NumPy scientific data analysis software package (Jones et al. 2012). Figure 4-5 provides a schematic of the rainfall excess production and excess routing components of the model. The gray boxes indicated the four main components of the model, plus the model output component. Source code for the model is provided in Appendix B. The model solves the infiltration, routing, and surface depression storage components of the model at each time step for every grid cell of the input parameters. The model output component deserializes model results into the NetCDF\(^{35}\) file format. NetCDF provides a standardized data model and format for multidimensional scientific data, supporting spatial and time

\(^{34}\) http://www.esri.com

\(^{35}\) http://www.unidata.ucar.edu/software/netcdf/
dimensions. NetCDF files are compatible with a variety of software visualization tools including ESRI’s ArcGIS® Desktop and ArcScene.

Data preparation steps involve a series of sub-models developed using ArcGIS® Model Builder. A brief description of the model inputs is provided here, detailed diagrams of the sub-models are available in Appendix A. The infiltration depth ($F_t$) and rainfall excess parameters are continuously updated during model execution. Generally, these parameters are set to zero at the start of model execution. Effective porosity ($VWC_{sat}$) is obtained from the section of this study on trend detection in effective soil porosity. Initial soil water content ($VWC_{initial}$) is obtained from measured values at the beginning of the modeled rainfall event or, in the case of simulations, estimated based on historical record. Saturated Hydraulic Conductivity ($K_{sat}$) is based on the spatial structure of saturated hydraulic conductivity section of this study. Wetting front suction ($\psi$) is estimated based on the procedure described by Rawls and Brakensiek (1982). The D-infinity flow direction grid is obtained using the TauDEM\textsuperscript{36} software. The digital elevation model parameter for the study region was derived from LIDAR elevation data from the U.S. Geological Survey\textsuperscript{37} and from Woolpert, Inc. survey drawings. Grid cell resolution is chosen to approximate flow velocity based on bioinfiltration cell cover type and average slope. For example, using the U.S. Department of Agriculture's upland method (US SCS 1972), the overland flow velocity for forest cover and 2% slope is 0.107

\textsuperscript{36} http://hydrology.usu.edu/taudem/taudem5.0/index.html

\textsuperscript{37} Data available from the U.S. Geological Survey
Figure 4-5. Rainfall excess production and flow routing model diagram.
m/s. If a model time step of one second is used, overland flow velocity dictates that the model cell size should be approximately 0.10m.

The Green-Ampt method captures heterogeneity across spatially varying parameters, e.g., $K_{sat}$, $\psi$, and $VWC_{sat}$. The Curve Number methodology takes an alternate approach based on empirically derived relationship between surface cover and runoff. The relationship has no direct basis in physical measurements of the soil. For this component of the model, impervious regions directly connected to bioinfiltration basins are assumed to have homogenous infiltration rates and higher surface velocities than the turf and vegetative cover types found within the basins. The impervious areas are lumped and described by a hydrograph according to the TR-55 method described by Chronshey (1986). The directly connected impervious regions are identified and delineated using the Watershed tool, which is part of the Hydrology toolkit of the ArcGIS® Spatial Analyst extension. Additional detail on this model component is provided in Appendix A-1.

Model Visualization

Model visualization is performed with the ESRI’s ArcScene® 3D visualization application. Elevation models for the native and media soils were generated based on contour datasets from surveyed drawings provided by Woolpert, Inc. Additional description of the methods used to interpolate survey contours to terrain models can be found in Appendix A-1. Raster surfaces were converted to triangulated irregular networks (TIN) and extruded into multi-patch volumes using the ESRI 3D Analyst toolkit. Cross sections were converted into 3D volumes using ESRI 3D Analyst. NetCDF model output
is loaded into ArcScene using the ESRI Multi-dimension Toolkit. The ‘Animation Manager’ is used to step through and record the model results.

Results and Discussion

Spatial Structure of Saturated Hydraulic Conductivity

Measured saturated hydraulic conductivity ($K_{sat}$) for the AOI ranged from 104.14 mm/hour to 1270.00 mm hr$^{-1}$ (Table 4-2). The value of 1270.00 mm hr$^{-1}$ far exceeded both the next highest observed value (579.12 mm hr$^{-1}$) as well as previous published findings for similar soil textures (e.g., 210.00 mm hr$^{-1}$ [Rawls et al. 1982]). High measurement values could be due by soil macropores or local scale processes present at the sample site. Even with additional proximal samples, it is difficult to ascertain the validity of this measurement without a corresponding understanding of local-scale processes present at the sampling location. The presence of a single large value with a small sample size ($n = 10$) makes estimation procedures based on the sample distribution problematic, the 1270.00 mm hr$^{-1}$ value was excluded in these circumstances.

The small sample size and high variation of the area of interest $K_{sat}$ measurements limited the statistical power available for analysis of spatial structure (Table 4-6). Exploratory data analysis indicates the presence of a positive skew (Figure 4-6) and necessitated a log transformation prior to further analysis. Transformed data approximates a normal distribution (Figure 4-7). The existence of spatial autocorrelation is tested using Moran’s $I$ with the log transformed $K_{sat}$ based on inverse Euclidean distance between samples. The Moran’s $I$ statistic indicates a lack of sufficient evidence to support the hypothesis of spatial autocorrelation in the $K_{sat}$ dataset ($I = -0.0018, p =$
### Table 4-6. Saturated hydraulic conductivity ($K_{sat}$) univariate analysis for native soil.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>10</td>
<td>9</td>
</tr>
<tr>
<td>Minimum (mm hr$^{-1}$)</td>
<td>104.14</td>
<td>104.14</td>
</tr>
<tr>
<td>Maximum (mm hr$^{-1}$)</td>
<td>1270.00</td>
<td>579.12</td>
</tr>
<tr>
<td>Mean (mm hr$^{-1}$)</td>
<td>451.87</td>
<td>360.96</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>321.98</td>
<td>153.85</td>
</tr>
<tr>
<td>Skewness</td>
<td>1.70</td>
<td>-0.12</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>5.37</td>
<td>2.11</td>
</tr>
<tr>
<td>1st Quartile (mm hr$^{-1}$)</td>
<td>274.32</td>
<td>259.08</td>
</tr>
<tr>
<td>Median (mm hr$^{-1}$)</td>
<td>396.24</td>
<td>365.76</td>
</tr>
<tr>
<td>3rd Quartile (mm hr$^{-1}$)</td>
<td>548.64</td>
<td>457.20</td>
</tr>
</tbody>
</table>

0.7019). Visual inspection of a semivariogram plot (Figure 4-8) supports the findings of the Moran’s $I$ test, indicating little to no spatial structure at the scales observed in this study. While the small sample size limited the ability of this analysis to distinguish spatial structure from noise, previous findings with larger sample sizes have produced similar results (Sobieraj et al. 2004).

The lack of spatial structure has implications for the parameter estimates of $K_{sat}$ required by local-scale infiltration models. It suggests that estimations based on nearby $K_{sat}$ measurements or geostatistical interpolation will not adequately capture the true variability of $K_{sat}$ at the local or point scale. In this case, it is more appropriate to estimate $K_{sat}$ based on a back-transformed value randomly selected from a lognormal $K_{sat}$ sample distribution.

**Volumetric Water Content and Rainfall Observation Data**

The study period included observations between June 1, 2011 and May 16, 2012. Precipitation data was available for the entire study period. Data collection for the volumetric water content ($VWC$) sites with intermittent, with only a subset of the sites
reporting for the entire study period. These issues were primarily due to power and network failures of the wireless sensor network and Internet connectivity problems.

The distributions of $VWC$ measurement for each sensor site tended to be positively skewed (Table 4-7 and Figure 4-9). Variations among the distribution varied both site-to-site (Figure 4-10 and 4-11) and depth-to-depth at the same site (Figure 4-11 and 4-12). As anticipated, $VWC$ measurements over the study period show a strong positive correlation with rainfall measurements. Figure 4-13 shows $VWC$ at multiple depths versus time for the study period with a hyetograph overlay. The monitoring site is located in the bioinfiltration basin located at the intersection of Chesterfield, Richland, and Park Streets. Figure 4-14 shows the same plot for an individual storm event, illustrating the rapid response of $VWC$ measurement to rainfall events. The $VWC$ response can reach saturation from antecedent conditions within a five-minute sampling interval. This point supports the use of higher frequency sampling rates, perhaps based on an adaptive sampling technique. Higher frequency monitoring allows a more accurate description of $VWC$ responses over time. This information is useful for tracking the position and movement of a wetting front in a soil column.
Figure 4-6. Histogram of measured \( (K_{sat}) \) for native soil (Source: measured \( K_{sat} \) from Schnabel Engineering, LLC, 2009).

Figure 4-7. Log normal QQ-Plot of measured saturated hydraulic conductivity \( (K_{sat}) \) (Source: measured \( K_{sat} \) from Schnabel Engineering, LLC, 2009).
Figure 4-8. Log transformed semivariogram for saturated hydraulic conductivity ($K_{sat}$) in native soils (Source: measured $K_{sat}$ from Schnabel Engineering, LLC, 2009).
Table 4-7. Effective soil porosity ($VWC_{sat}$) univariate analysis for all monitoring sites.

<table>
<thead>
<tr>
<th>Statistic</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>1285571</td>
</tr>
<tr>
<td>Minimum ($m^3m^{-3}$)</td>
<td>0.1001</td>
</tr>
<tr>
<td>Maximum ($m^3m^{-3}$)</td>
<td>0.6499</td>
</tr>
<tr>
<td>Mean ($m^3m^{-3}$)</td>
<td>0.2182</td>
</tr>
<tr>
<td>Standard Deviation</td>
<td>0.0868</td>
</tr>
<tr>
<td>Skewness</td>
<td>1.3388</td>
</tr>
<tr>
<td>Kurtosis</td>
<td>4.8105</td>
</tr>
<tr>
<td>1st Quartile ($m^3m^{-3}$)</td>
<td>0.1595</td>
</tr>
<tr>
<td>Median ($m^3m^{-3}$)</td>
<td>0.1886</td>
</tr>
<tr>
<td>3rd Quartile ($m^3m^{-3}$)</td>
<td>0.2738</td>
</tr>
</tbody>
</table>
Figure 4-9. Histogram for measured volumetric water content at all sites and depths between June 2011 and May 2012.

Figure 4-10. Histogram for measured volumetric water content at site CRPS site 1, depth 12” between June 2011 and May 2012.
Figure 4-11. Histogram for measured volumetric water content at site CRPS site 7, depth 12” between June 2011 and May 2012.

Figure 4-12. Histogram for measured volumetric water content at site CRPS site 7, depth 18” between June 2011 and May 2012.
Figure 4-13. Precipitation and soil water content for CRPS site 4 at multiple depths between August 2011 and May 2012.
Figure 4-14. Precipitation and soil water content for CRPS site 4 at multiple depths between September 21, 2011 and September 24, 2011.
Trend Detection in Effective Volumetric Water Content at Saturation

Exploratory data analysis indicated a slight positive skew in the distribution of saturated volumetric water content ($VWC_{sat}$) observations (Figure 4-15), but not enough to warrant a data transformation. The relationship between observed $VWC_{sat}$ and time (days) was visually evaluated for each site and depth (Figure 4-16). A negative trend is evident in several of the sensor sites. The 95% confidence interval plot (Figure 4-17) indicates non-zero slope terms at individual sensor sites, further supporting the presence of an identifiable trend. A linear model was fit to the entire data set with $VWC_{sat}$ as the dependent variable and time (days since beginning of monitoring) as the independent variable. The resulting linear model shows a statistically significant slope with $b = -0.0004238$, $t (226) = -4.54$, $p < 0.0001$ (Figure 4-18). Visual inspection of residuals vs. fitted values (Figure 4-19) indicates that no violation of homoscedasticity. A normal probability plot of fitted residuals indicates that the assumption of normality is valid (Figure 4-20).

A linear mixed effects model is applied to the $VWC_{sat}$ dataset to evaluate the influence of time, site, media, and depth on observed values. The initial model incorporated the fixed effects of time, media, depth, and the random effect of site. Iterative removal of effects and examination of model results indicate that media did not
Figure 4-15. Histogram of effective soil porosity ($VWC_{sat}$) without log transformation.

Figure 4-16. Effective soil porosity ($VWC_{sat}$) versus time linear relationship by site and depth.
**Figure 4-17.** Confidence intervals for effective soil porosity ($VWC_{sat}$) versus time linear relationship by site and depth.

**Figure 4-18.** Linear model for effective soil porosity ($VWC_{sat}$) versus time.
**Figure 4-19.** Residuals plot for effective soil porosity ($VWC_{sat}$) versus time.

**Figure 4-20.** Normal probability plot for residuals of effective soil porosity ($VWC_{sat}$) versus time linear model.
significantly improve the model. Subsequently, the final model included only time and depth as fixed effects and site as a random effect. The exclusion of the media effect from the model may have resulted from redundancy between the depth and media effects. The sensor sites observing native soil were located at depths of 20.32 cm (8 in) and 45.72 cm (18 in), while the sensor sites observing engineered media were located at depths of 15.24 cm (6 in) or 30.48 cm (12 in).

As with the linear model, the time effect was found to be significant with $F(1, 216) = 14.00, p = 0.0002$. The depth effect was significant with $F(1, 216) = 4.47, p = 0.0356$. No significant interaction between effects was detected at a 95% significance level. The fixed effects model was compared with a model containing only random effects using a likelihood ratio test with $p < 0.0001$, indicating that the mixed effects approach offered an improvement over a null model. The mixed effects model produced a negative coefficient estimate for the depth effect ($b = -0.0037$), indicating that $VW_{C_{sat}}$ observations at deeper depths experience a greater decline in moisture holding capacity over time.

Results from this analysis indicate a negative trend in $VW_{C_{sat}}$ over time at the sensor sites in Aiken, SC. A declining trend in effective porosity has implications for the performance of bioinfiltration basins. The basin located near the intersection of Park, Union and Fairfield Street (PUF) is used to illustrate the impact of trends in effective porosity. The engineered media located within the PUF basin occupies a volume of 198.87 m$^3$. Assuming an effective porosity of 0.42 m$^3$m$^{-3}$ at the start of the study, the engineered media located in PUF would have the capacity to store 82.93 m$^3$ of water.
fully saturated. Application of the linear model produces an estimated porosity of 0.32 m$^3$m$^{-3}$ yielding a storage capacity of 64.24 m$^3$, a reduction of 18.70 m$^3$. A reduction in effective porosity decreases the time to ponding, thus lowering the total storm water handling capacity during a given time period. The decline trend in observed VWC$_{sat}$ during the study period may have resulted from factors other than a decline in effective porosity. There is an underlying assumption in this analysis that the observed VWC$_{sat}$ was obtained under truly saturating conditions. A closer investigation of the first and second rainfall events indicates a potential violation of this assumption. The two rainfall events occurred just 24 days apart with the second event resulting in significantly higher observed VWC$_{sat}$; one tailed paired $t$(20) = -5.12, $p < 0.0001$. This contradicts the overall findings and indicates that other factors are influencing observed VWC$_{sat}$. One explanation is a variation in rainfall intensity and magnitude between the events. The first event was a short duration storm with 1.6 cm (0.63 in) of rainfall. The second event occurred over the course of 24 hours with 11.30 cm (4.45 cm) of rainfall and resulted in a greater ponding depth and duration. The prolonged saturation of the media may have produced higher valued VWC$_{sat}$ observations. While consistency in ponding depth and duration is unlikely to occur in an observational study, a direct ponding depth measurement would have aided an assessment of the relationship between ponding depth and duration on observed VWC$_{sat}$. The only depth measurements available in this study were taken inside of basin outflow structures and proved to be a poor indicator of ponding depth due to the introduction of water into the structure from sources besides the basin and the high threshold required for water to “overflow” into the structure. In
addition to ponding depth measurements, future studies would benefit from the inclusion of an antecedent condition effect into the model, perhaps by introducing a soil moisture deficit term ($VWC_{sat} - VWC_{residual}$).

**Runoff and Infiltration Production and Routing Model**

In order to evaluate the bioinfiltration model, a sample rain event from the monitoring period was chosen. During the study period, five bioinfiltration basins underwent data collection between 2010 and 2012. For this analysis, the CRPS basin (Figure 4-1) was chosen as a representative bioinfiltration basin. This basin is located near the intersection of Chesterfield, Richland, and Park streets. It contains four soil moisture-monitoring assemblies at depths described in Table 4-5. CRPS has three curb cuts directly connected to impervious regions including roadways and parking areas. Rainfall excess originating from the northeastern curb cut was significant enough to cause rill erosion between the curb cut and the bioinfiltration basin. A rainfall event producing 6.375 cm (2.51 in) of rainfall occurred on September 21, 2011. Event rainfall and initial soil water content ($VWC_{sat}$) from prior to the start time was used to parameterize the model. The remaining parameters were selected based on the steps described in the methods section of this study. A cell size of 0.10 meter was used for all raster inputs. The time step was set to one second, with model output serialized every fifteen seconds.

Results from the model for the September 21, 2011 rainfall for an individual grid cell located at CRPS Site #4 are shown in Figure 4-21. The infiltration rate is equivalent to the rainfall rate until $t_p$ is reached. The time to ponding cannot be directly calculated
Figure 4-21. Simulation of Sept. 21, 2011 rainfall event for CRPS #4.

for unsteady rainfall, but can be approximated by the infiltration rate curve. Because the rainfall rate was below the mean infiltration rate (Table 4-6), no significant surface runoff from the soil media and surrounding native soil occurred.

The performance of the bioinfiltration model is evaluated by comparing the time for the wetting front to reach the depth of the soil water content sensors between the model and the measured response. This approach is limited by the assumptions implicit in the Green-Ampt equations, namely that a sharp wetting front exists and full saturation occurs. Saturation ($VWC_{sat}$) is based on the point where $VWC$ measurements cease to increase in a saturating rain event. However, this validation step requires an accurate saturation time. The accuracy of the observed saturation time is limited by the five-minute sampling interval. This process can be evaluated at each site and depth where information is available. For the CRPS cell, a single rain event was simulated and compared against observed measurements. The modeled time to wetting front were found
to follow the observed times at depths for locations CRPS Site 1, 2 and 4 (Site 3 was offline during rainfall event). The Nash-Sutcliffe Efficiency (NSE) results indicate a model performance of 0.94 ($n = 7$). A value of 1 indicates a perfect correspondence to the observations. A value of zero indicates the predictions are as accurate as the mean of observed values. The small number of observed versus simulated data points weakens the strength of this NSE metric. More rain events would strengthen this model performance measure. A number of model factors contribute to this model efficiency. The uncertainty associated with the estimation of the soil parameters, particularly saturated hydraulic conductivity, result in discrepancies between observed and simulated results. The contribution of directly connected impervious is also subject to uncertainty related to area estimations. Runoff in urban areas is difficult to accurately quantify and flow contribution area may change due to any of a number of dynamic factors including vehicle traffic and storm sewer failures in other parts of the watershed. This study did not incorporate surface ponding level measurements, which would benefit future validation tests of this model. Future revisions of the model might incorporate vegetation interception, evaporation, and transpiration components to better capture the effective inputs of rainfall on the basin and surrounding areas. Additionally, this model assumes a constant overland flow velocity for sheet and shallow concentrated flow. The small spatial extent of the watersheds used for this study make a constant flow velocity assumption reasonable. Over larger areas, greater flow channelization is expected, leading to a wider range of possible velocities. In these cases, a kinematic wave solution
for overland flow would provide a more realistic description of flow velocity by incorporating parameters of flow depth, Manning’s roughness and slope.

**Model Visualization**

Model visualization is performed on the CRPS cell described above. A simulated rainfall event is used in order to cause greater ponding depths to occur within the basin depression. The U.S. National Weather Service Precipitation Frequency Data Server (PFDS)\(^{38}\) was used to determine a 50-year precipitation frequency estimate for Aiken, SC. A rainfall event of 63.00 mm (2.48 in) for a duration of 30 minutes was used for the model simulation. Rainfall was allocated so that a 50-year, 5-minute rainfall occurred beginning at a time of ten minutes. Initial volumetric water content was chosen to be 0.22 m\(^3\) m\(^{-3}\). A grid cell size of 0.1 m was used with a time step of one second. Model output was written at fifteen-second intervals.

A two dimensional view of surface excess and basin ponding at three different time steps is shown in Figure 4-22, with a heat map indicating rainfall excess depth. The lighter blue areas denote areas where surface excess is occurring, and hotter colors indicate areas where water is channelizing (e.g., from curb cuts) or is accumulating in basin depressions. Figure 4-23 shows a cross sectional view of saturation depth in addition to surface excess. The graph in Figure 4-24 shows saturation depth, infiltration depth and excess depth over time for the same rain event at Site 1. Figure 4-25 and 4-26 show the same cross sectional view and graph with an adjusted effective porosity measurement. The impact of the adjustment is visible between Figure 4-23 and 4-25. The

\(^{38}\) [http://www.nws.noaa.gov/oh/hdsc/index.html](http://www.nws.noaa.gov/oh/hdsc/index.html)
visualizations show that media and native soils with high infiltration rates can support large volume of storm water storage. Based on these simulations it appears that the basins in the City of Aiken study region could support higher volumes of storm water inflow, perhaps through additional storm sewer pipe connections to the basin.

Conclusions

LID design philosophies are increasingly guiding stormwater BMPs, with bioinfiltration systems playing a key role in these management techniques. The benefits of bioinfiltration systems are many, with research documenting reductions in peak flows, increases in ground water recharge, decreases in runoff volumes, and pollutant filtering. An improved understanding of the processes that govern the efficacy of bioinfiltration BMPs will encourage wider adoption of LID approaches. This research incorporated GIS technology to characterize and model bioinfiltration systems at scales capable of capturing the complex biological, pedological, and hydrological processes that govern their performance. A soil chemical and physical property analysis found that soils in the region identified by this study were characterized by high infiltration rates, allowing naturally high rates of storm water infiltration. This illustrates the benefit of adopting pervious cover types within the City of Aiken, allowing natural processes to lessen storm water volumes. The high standard deviation of infiltration rates suggests that local scale processes (e.g., biological activity) play a significant influence on the hydraulic properties of soil. Analysis of the spatial structure of infiltration measurements in the study region reinforces existing findings that soils exhibit limited spatial dependency at
Figure 4.22. CRPS. 50 year, 30 minute rainfall simulation.
Figure 4-23. CRPS. 50 year, 30 minute rainfall simulation without effective porosity adjustment. Time: 30 minutes.
Figure 4-24. CRPS site 1. 50 year, 30 minute rainfall simulation without effective porosity adjustment.
Figure 4-25. CRPS. 50 year, 30 minute rainfall simulation with effective porosity adjustment. Time: 30 minutes.
Figure 4.26. CRPS site 1. 50-year, 30 minute rainfall simulation with effective porosity adjustment.
the scales influencing urban stormwater BMPs. The lack of spatial structure has implications for hydrologic modeling applications that depend on soil property estimation.

Long term monitoring of bioinfiltration systems is important to refining engineering design practices and creating recommendations for maintenance strategies. This research demonstrated the utility of WSN technology as a means to implement continuous monitoring. Long-term data can identify temporal trends in infiltration performance. This investigation found that observed effective soil porosity for the monitored bioinfiltration basins experienced gradual decline over time. This finding suggests that bioinfiltration systems may lose subsurface stormwater storage capacity over time. The use of effective soil porosity as a bioinfiltration performance indicator may help guide future decisions regarding engineered media composition.

Modeling is an important component of stormwater BMP design. Conventional urban stormwater design tools focus on point-scale or catchment scale processes. As a local or in-situ control measure, bioinfiltration systems occupy a problem space that is not encompassed by these scales. A grid-based within catchment scale model is described that operates at scales fine enough to capture the spatial heterogeneity of processes affecting bioinfiltration operation. The model is suited to evaluating the significance of variations in soil hydraulic properties, including those resulting from temporal trends. In addition to quantitative results, the model is conducive to qualitative interpretation through 3D visualization. Hypothetical bioinfiltration system performance was simulated using a 50-year rainfall event.
Acknowledgements

Authors wish to acknowledge the Clemson University Center for Watershed Excellence for their support of this research; the City of Aiken for their support through funding, engineering guidance, and information technology support; and Woolpert, Inc and Schnabel Engineering, LLC for sharing of engineering technical reports.

References Cited


Saxton, K. E., and W. J. Rawls. 2006. “Soil Water Characteristic Estimates by Texture 
and Organic Matter for Hydrologic Solutions.” Soil Science Society of America 
Journal 70 (5): 1569.

Watersheds, ed. C.T. Haan, H.P. Johnson, and D.L. Brakensiek, St. Joseph, 
Michigan: ASAE Monograph No 5. 121–166.


South Carolina Department of Natural Resources. 2011. South Carolina Climate Atlas. 

Stankowski, S. J. 1972. “Population Density as An Indirect Indicator of Urban and 
800-B: B219–B224.

Tarboton, David G. 1997. “A New Method for the Determination of Flow Directions and 
Upslope Areas in Grid Digital Elevation Models.” Water Resources Research 33 

of engineered soil media for bioretention basins.” Transactions of the ASAE 51 (2): 
499-514.


Through Low Impact Development (LID) Strategies and Practices, EPA 841-F-07- 
006.

APPENDICES
Appendix A

ArcGIS® Models for Bioinfiltration Analysis

Figure A-1. ArcGIS® model for directly connected impervious areas.
Figure A-2. ArcGIS® model for initial soil water content ($VWC_{initial}$).
Figure A-3. ArcGIS® model for soil effective porosity ($VW_{C_{sat}}$).
Figure A-4. ArcGIS ® model for saturated hydraulic conductivity ($K_{sat}$).
Figure A-5. ArcGIS® model for Green-Ampt wetting front suction ($\theta$).
Figure A-6. ArcGIS® model for D-Infinity flow direction.
Appendix B

Bioinfiltration Model Source Code

Table B – 1. GreenAmpt.py

```python
"""
GreenAmpt.py
---------
Green-Ampt (1911) infiltration and rainfall excess model.

Notes:
-----
"""
__author__ = "Samuel T. Esswein"
__copyright__ = "Copyright 2012"

import sys
from time import clock, gmtime
from numpy import *

class GreenAmpt:
    """ Green-Ampt infiltration and rainfall excess model."""

    stepSize = 60

    def __init__(self, k, psi, deltaTheta, basinStorage, flowRoute, dcia):
        """ Parameters must be of type ndarray."""

        self.yDim, self.xDim = yDim, xDim = k.shape
        self.k = k
        self.dcia = dcia
        self.basinStorage = basinStorage
        self.flowRoute = flowRoute
        self.kPsiDeltaTheta = k * psi * deltaTheta
        self.deltaTheta = deltaTheta
        self.mask = (k > 0)
        self.excess = zeros((yDim, xDim))   # Rainfall excess initial condition
        self.fDepth = zeros((yDim, xDim))   # Infiltration depth initial condition

    def runModel(self, precipTS, nExcess = None, nFRate = None, nFDepth = None,
                 nZDepth = None, nPrecip = None, writeIncrement=60):
```
"""Args:
    n*: isntance of Dataset.Netcdf
writeIncrement: Time increment (s) to serialize to NetCDF.
"""

excess = self.excess
fDepth = self.fDepth
steps = len(precipTS)

if nExcess is not None: nExcess.append(precipTS[0,0], excess)
if nFDepth is not None: nFDepth.append(precipTS[0,0], fDepth)
t0 = clock()

for t_min in xrange(0, steps):
dciaDepth = self.dcia.calculateDepthByArea(t_min)/self.stepSize
precip = precipTS[t_min, 1] / self.stepSize * self.mask

t1 = clock()
sys.stdout.write("Step %d of %d.\n   Status: " % (t_min + 1, steps))

for t_sec in xrange(0, self.stepSize):
cellInput = precip + excess + dciaDepth;
fRate = self.kPsiDeltaTheta / (fDepth + 0.00000000001) + self.k
fDepth = minimum(cellInput, fRate) + fDepth;
excess = (cellInput - fRate).clip(min=0);

if any(excess):
excess = self.flowRoute.route(excess);

if self.basinStorage is not None:
excess = self.basinStorage.ponding(excess)

if ((t_min*60 + t_sec) % writeIncrement == 0):
    secondsSince = precipTS[t_min,0] + t_sec

    if nExcess is not None: nExcess.append(secondsSince, excess)
    if nFRate is not None:  nFRate.append(secondsSince, fRate)
    if nFDepth is not None: nFDepth.append(secondsSince, fDepth)
    if nZDepth is not None:
zDepth = fDepth / self.deltaTheta
nZDepth.append(secondsSince, zDepth)
if nPrecip is not None: nPrecip.append(secondsSince, precip)

sys.stdout.write("n Complete (%fs)\n % (clock() - t1))
sys.stdout.write("nModel Run Complete. Elapsed Time: %f\n % (clock() - t0))

Table B – 2. FlowRoute.py

"""
FlowRoute.py
----------
Routes accumulated rainfall excess based on grid-based flow direction algorithm.

Classes:
-------
D8 - Single direction flow routing. Transfers all accumulated excess to one neighboring cell.

D-infinity - Multiple direction flow routing. Proportions accumulated excess to one or two neighboring cells based on the angle of steepest descent out of the cell. Modification of algorithm described by Tarboton (1997).

References:
----------
"""
__author__ = "Samuel T. Esswein"
__copyright__ = "Copyright 2012"

from numpy import *

class Dinfinity:
    """D-Infinity Algorithm described by Tarboton (1997)."""
    pi4 = pi / 4
    def __init__(self, dinf, dem):
Args:
    Dinf - Flow direction raster with radian angles.
    Dem - Filled and prepped elevation model.

    self.yDim, self.xDim = yDim, xDim = dinf.shape
    self.offset = [1, -xDim + 1, -xDim, -xDim - 1, \
                   -1, xDim - 1, xDim, xDim + 1]
    self.dinf = dinf.ravel()
    self.sdem = argsort(dem, axis=None)
    self.cells = len(self.sdem)

def route(self, exc):
    """Routes accumulated excess rainfall once for each cell."""

e = exc.ravel()
rExc = zeros((self.cells))

for idx in self.sdem:
    angle = self.dinf[idx]

    if (e[idx] > 0): # Is there excess to route?
        if not isnan(angle):
            quad = self.prop(angle)
            pos1 = idx + self.offset[quad[0]]
            pos2 = idx + self.offset[quad[2]]
            prop1 = quad[1] * e[idx]
            prop2 = quad[3] * e[idx]
            rExc[pos1] += prop1
            rExc[pos2] += prop2
        else:
            rExc[idx] = e[idx]

rExc.resize(self.yDim, self.xDim)
return rExc

def prop(self, angle):
    """
    Proportions accumulated excess between one or two cell neighbors
    based on a flow direction angle specified in radians.
    """
    qr = divmod(angle, self.pi4)
    prop = qr[1] / self.pi4
    idx = int(qr[0])
return (idx, 1 - prop, (idx + 1) % 8, prop)

Table B – 3. FlowRoute.py

```
BasinStorage.py
--------------
Distributes accumulated ponding volume across surface depression.

Notes:
------

__author__ = "Samuel T. Esswein"
__copyright__ = "Copyright 2012"

import sys
from time import clock
from numpy import *

class BasinStorage:
    ""
    Relates basin elevation to storage volume. Supports redistribution of
    accumulated ponding volume across a surface depression.
    ""
    
    def __init__(self, dem):
        """Args: digital elevation model (dem) is ndarray."""
        
        self.sDemIdx = argsort(dem, axis=None)
        sDem = dem.ravel()[:, self.sDemIdx]
        self.rsDem = sDem - sDem[0]
        self.crsdem = cumsum(self.rsDem)
        self.yDim, self.xDim = dem.shape

    def ponding(self, excess):
        exc = excess.ravel()
        csExc = exc[:, self.sDemIdx].cumsum()
```
for pos in range(0, self.sDemIdx.size-1):
    pondingElev = (csExc[pos] + self.crsdem[pos]) / (pos+1)

    if pondingElev <= self.rsDem[pos]:
        break

exc[[self.sDemIdx[0:pos+1]]] = pondingElev - self.rsDem[0:pos+1]
exc.resize(self.yDim, self.xDim)
return exc