Conservation Planning Support System
Software Architecture Document

Version <1.1>
### Revision History

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;05/15/2003&gt;</td>
<td>&lt;1.0&gt;</td>
<td>First Draft of the Software Architecture Doc.</td>
</tr>
<tr>
<td>&lt;12/04/2003&gt;</td>
<td>&lt;1.1&gt;</td>
<td>This revised version contains a more complete portion for each of the design states.</td>
</tr>
</tbody>
</table>
Table of Contents

1. Introduction 4
   1.1 Purpose 4
   1.2 Scope 4
   1.3 Definitions, Acronyms, and Abbreviations 5
   1.4 References 5
   1.5 Overview 5

2. Architectural Goals and Constraints 5

3. Architectural Representation 5
   3.1 Architectural Views 5
   3.2 Architectural Design Patterns 6
   3.3 Architectural Style 6
   3.4 Architectural Process 7

4. Architectural View Decomposition 8
   4.1 Use-Case View 8
      4.1.1 Use-Case Realizations 9
      Formulate Alternatives 9
      Evaluate Alternatives 9
      Make Decision 9
   4.2 Design View 9
      4.2.1 Overview 10
   4.3 Process View 10
   4.4 Component View 10
      4.4.1 Overview 10
   4.5 Deployment View 10

5. Size and Performance 10

6. Quality 11

7. Bibliography 11
1. Introduction

This introduction provides an overview of the entire Software Architecture Document for the Conservation Planning Support System. It includes the purpose, scope, definitions, acronyms, abbreviations, references, and overview of the system.

1.1 Purpose

This document provides an architectural overview of the Conservation Planning Support System (CPSS). The primary purpose of the Conservation Planning Support System (CPSS) is to satisfy the agricultural needs of producers and soil conservationists.

This document is intended to capture and convey the significant architectural decisions which have been made in designing and building the system. It is a way by which the systems’ architect and others involved in the project can better understand the problems to be solved and how it will be represented with this system.

The following diagram shows the different steps that will be taken to set and get information from the MARIA database effort:

1.2 Scope

The scope of this document is the implementation of the CPSS which consist of steps 1 through 9 of the NRCS National Planning Process.
1.3 Definitions, Acronyms, and Abbreviations

See the System Glossary.

1.4 References

As explained below, the top-level composition of the Conservation Planning Support System, such as the use case models, uses the concepts described in the books: Advanced Use Case Modeling; Frank Armour and Granville Miller.

The reference book used to model all the architectural diagrams was: The Unified Modeling Language Reference Manual; James Rumbaugh, Ivar Jacobson, and Grady Booch.

Some concepts regarding the documentation of software architecture were obtained from the book: Documenting Software Architecture-Views and Beyond; Paul Clemens, Felix Bachmann, Len Bass, David Garlan, James Ivers, Reed Little, Robert Nord, and Judith Stafford.

1.5 Overview

This document consists of 7 sections, which are described below:

• Section 1 is simply an introduction to the software architecture of the CPSS.
• Section 2 addresses the goals and constraints of the system’s architecture
• Section 3 describes the architectural representation of the system.
• Section 4 describes the five views in which the system documentation is divided by following the Rational Unified Process (RUP). A visual representation of these five views can be seen in the Software System Development Diagram.
• Section 5 of this document talks about other system considerations such as size and performance of the system.
• Section 6 describes some system quality issues.
• Section 7 is a bibliography of the references used to create this document.

2. Architectural Goals and Constraints

The CPSS architecture has been designed with the following objectives in mind:

1. To facilitate the decision making process for producers by incorporating information on sustainability, offsite effects, and profitability.
2. To help soil conservationists explain how management changes affect the natural system’s production, sustainability, and water quality effects.
3. To allow producers to enhance their agricultural planning by giving them a logical decision based on available science.
4. To allow producers to become more profitable from their crops since a better plan will have been implemented.

The major design and implementation constraints for the system are:

1. Simplicity
2. Flexibility

A full list of the system requirements can be seen in the High Level Requirements document.

3. Architectural Representation

3.1 Architectural Views

Modeling, implementing, and documenting a system requires that the system be viewed from different perspectives. Because of this, the architecture of the CPSS will be represented in a five view approach: Use Case View, Design View, Process View, Component View, and Deployment View. The following is a brief description for each of the views:
Use Case View: the main purpose of the use case view is to define main drivers of the system, which are the system requirements.

Design View: this view contains any system definitions as well as class and object diagrams which depict the services that the system will provide to its end-users.

Process View: this view will display the processes that form the systems’ mechanism. These will be represented as collaboration, sequence, and activity diagrams.

Component View: this view will include system and user interface specifications, meaning, the different components that make up the system.

Deployment View: this view will depict how the different systems’ hardware nodes will come to life together as well as how each of the hardware nodes will be installed and deployed.

As seen in the above diagram, requirements, or the use case view, are the main driving force to this and any software architecture system.

A more detailed map to these five models can be found in the Software System Development Diagram.

### 3.2 Architectural Design Patterns

The design pattern used to create the CPSS is the MVC (Model View Controller) design pattern. The MVC design pattern clearly separated the web application’s behavior, presentation and control. The modularity of this design pattern allows for easier code reuse, more centralized control, bugs easier to track down and code easier to modify. The presentation, or view, of the CPSS has been implemented keeping in mind the Model 2 usage pattern, which makes use of servlets as front controllers and maps incoming request to specific operation and selects views based on the model and session state. In some steps of the application the Model 1 pattern, in which a servlets is not used as the front controller, is also used. Some of the sections of the CPSS where Model 1 architecture is used are the DB Input For and the CPPE tool. These tools are very simple as far as navigation and a front controller is not needed since the controller is implied through the higher level container (the CPSS).

### 3.3 Architectural Style

As with any other things, a style may be used to satisfy any functional, non-functional or aesthetic needs in a software system. The CPSS, in particular, follows the three-tier architectural style: presentation tier, business tier, and data tier. The following is a simple description of what will be included in each of the tiers:
3.4 Architectural Process

CPSS follows the Rational Unified Process (RUP), whose goal is to enable the production of the highest quality software that meets end user needs within a predictable schedule and budget. The RUP is an iterative process divided into four phases: inception (the establishment of the projects business case), elaboration (establishment of project plan and system architecture), construction (the system implementation), and transition (system deployment).

During the inception phase, the business case and vision documents will be defined for the CPSS. The business case includes the criteria needed for the successful development of the system, a budget estimate of the resources needed to complete the system, and a schedule of the major milestones. The vision document defines the project’s problem to be solved, the project’s purpose and the project’s major stakeholders. During this phase, a PowerPoint presentation for the entire system will be created which will serve as a proof-of-concept for the system. This presentation will be reviewed by the major stakeholders for soundness and feasibility.

During the elaboration phase, the CPSS will have an easy to understand software architectural foundation and an implemented project plan (this was done first). As previously described, the architectural foundation is composed of a set of UML diagrams that entirely describes the system’s functionality.
During the construction phase, the CPPS will be incrementally and iteratively developed and tested. The development implies the complete coverage of the software requirements by following the different UML diagrams defined in the previous phase. At the end of this phase, the CPSS should be a completely designed, implemented, and tested system that is ready to be deployed for use.

Deployment of the CPSS will occur during the transition phase where the system will be placed on a server for the use of producers, soil conservationists, and database administrators. Once the beta release of the CPSS has been released, any system issues that arise will be logged and corrected and a new system version will be released. This process will continue until all new and legacy system requirements have been satisfied.

### 4. Architectural View Decomposition

#### 4.1 Use-Case View

The general functionality of the CPSS can be seen in the following diagram:

The CPSS will implement the architecturally-significant subset of the use cases specified above, which
include the items in Phase II. These use cases are described in the following section.

All implemented use cases have been associated in the Step 5 Base Use Cases.doc.

4.1.1 Use-Case Realizations

This section illustrates how the software actually works by giving an example on how some use-cases will be used within the system. This section will also examine how the various design structures of the CPPE contribute to its functionality.

Under Development!

4.1.1.1 Architecturally Significant Use Cases

The architecturally-significant use cases are those that “exercise” the most critical parts of the system architecture and demonstrate the core system functionality. In the case of the CPSS, the architecturally significant use cases that the ARS will be developing will be those that exist in Phase II of the NPPH:

- Formulate Alternatives: shows which management alternatives are available for the producer
- Evaluate Alternatives: shows a set of practical and evaluated CMS alternatives for the producer
- Make Decision: rank alternatives and select one to implement

These and other sub-use cases are shown and briefly described below:

Formulate Alternatives
Brief Description: The Soil Conservationist develops alternatives that will achieve the objectives of the client, solve the identified problems, take advantage of opportunities, and prevent additional problems from occurring.

Evaluate Alternatives
Brief Description: The effects of each alternative are evaluated and the impacts are described.

Make Decision
Brief Description: The producer determines which alternative(s) to implement and the Soil Conservationist prepares the necessary documentation.

4.2 Design View

This section describes the logical structure of the system as well as any definitions or acronyms. It starts from the overview of the architecture and then presents its key structural and behavioral elements such as usage and dependency. The documents that make up the logical view for the CPSS are:
4.2.1 **Overview**
This subsection will describe the overall decomposition of the CPSS design model as well as the interactivity that happens between components.

4.3 **Process View**
The process view will describe the system’s decomposition as well as the forms of communication between processes, like message passing, activity between components, and message sequencing.
The documents that are incorporated into this view are:

- Collaboration Diagram.doc
- Sequence Diagram.doc
- Activity Diagram.doc

4.4 **Component View**
The component view will describe the overall component and subsystem organization of the CPSS. This document will reside in the Component View folder.
The documents included in this view are the following:

- User Interface Specifications.doc
- System Interface Specifications.doc
- Software Architecture Documentation.doc (this document)

4.4.1 **Overview**
This subsection will include the various software layers that will exist in the system as well as the boundaries between these layers.

4.5 **Deployment View**
The CPSS deployment view of a system shows the physical nodes on which it executes. The CPSS is comprised of three physical nodes: the browser, the application server, and the database server. The simplicity of the CPSS physical view can be seen in the diagram below:

5. **Size and Performance**
The CPSS is far from being finished but the size of the system can be pre-summarized as follows:

- Number of use cases implemented: 9
6. **Quality**

System quality issues or concerns will go here.

7. **Bibliography**


