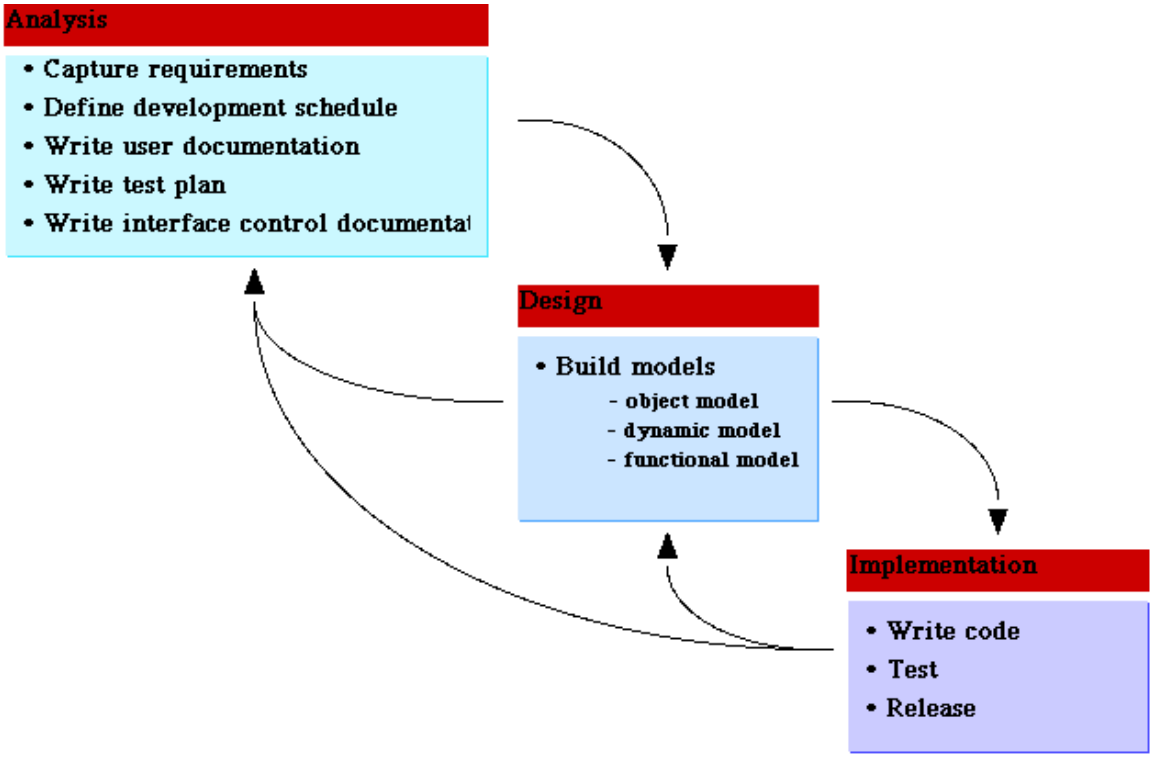
SLS Software Development Plan

Introduction

This document specifies the procedures to be used to produce software for the SLS control system. The process to be followed for developing software will follow a number of stages:



These stages will be iterative, new information discovered at any stage, may make it necessary to jump back to a previous stage and make modifications. Those changes will then iterate down through the intermediate stages, until it is possible to continue. For instance while writing the Man page for an application, it may be discovered that the requirements document is not clear on some point. The Requirements document and the development schedule would then be updated to reflect this, and writing the user documentation can then continue.

Documentation

All 'user' documentation will be presented in three forms from a common source.

- Unix man page
- Windows help file
- HTML file

The common source will (probably) be the HTML file, based on a standard template. Tools (html2man / html2help?) will need to be selected to help with the conversion. Each developer is free to select his favorite tools/packages to help build the HTML file, as long as it handles the sls template. Such tools might include packages such as framemaker or the netscape html editor.

Style

All source code will be commented in the following manner :

- Each source file will contain a revision history: generated from rcs - using the "id directive.
- Each function/method will include a description of each of its parameters and return value.
- Each variable will have a description of its use
- Each lexical level will be indented by one tab char
- C programs will use ANSI C declarations

Languages

Java and Tcl will be the languages used: but some 'system' code will use c++/c. Vesions used will be (subject to revision):

- Java : jdk1.1
- Tcl/Tk/tclX/blt: 8.0
- c/c++ : g++ 2.7.2
- make : gnumake 3.74

Source Control

CVS will be used as the mechanism for controlling and releasing development code versions. All source code must be checked in when not in use so that others can check it out as necessary. Source code for a project should be built in the project directory, not under a personal home directory.

Directory structure

Each directory containing source or executable file must include a README.txt file which lists each file in the directory as well as a short description of its function. The directory structure of the control system code will follow the class hierarchy of the project.

User/Group Identifiers

Each person who is developing software or maintaining the control system, will have their own user/ID and password. No global or shared accounts will be used. Developers will be allocated to one or more Groups depending on the functions they might be carrying out. File masks must be set so that object and executable files can be overwritten by other members of the group building a new version. Developers will use newgrp to gain appropriate access rights when working on different projects. default group will be sls - 102, but controls development will be done under 'controls' group.

Testing

Testing will be verified by a person other than the programmer. The tester will first verify that the testing plan meets the functional description contained within the user man page. The report shall include:

- Modifications to and additional information required in the testing plan.

- Conformance to Style Guide
- Results of the tests.
- Changes Required
- Improvements Suggested