

'Component Name'
Document Number: 'Document Number'

Software Design Document for 'Component Name'

Version: 'Version Number'

'Organization'

'Date'

Prepared by:
Wanda L. Reid

Table of Contents

4	Component Detail Design.....	3
4.1	Module Design.....	3
4.1.1	Module 1	3
4.1.1.1	Processing	3
4.1.1.2	Interfaces	3
4.1.1.3	Dependencies.....	3
4.1.1.4	Resources	4
4.1.1.5	Data	4
4.1.1.6	Submodules.....	4
4.1.1.6.1	Submodule 1	4
4.1.1.6.1.1	Interface Design.....	4
4.1.1.6.1.2	<i>Algorithm (PDL)</i>	4
4.1.1.6.1.3	<i>Restrictions/Limitations</i>	5
4.1.1.6.1.4	<i>Local Data Structures</i>	5
4.1.1.6.1.5	<i>Performance Issues</i>	5
4.1.1.6.2	Submodule 2	5
4.1.2	Module 2	5

4 Component Detail Design

This section gives a detail description for 'component name' by decomposing it into individual design modules and describing the significant properties and relationships among the modules. Each module is described by a set of attributes, which provides all information needed for implementation.

4.1 Module Design

This section specifies the attributes for each module. The attributes include: name, type, function, purpose, processing, interfaces, dependencies, resources, data and submodules.

4.1.1 Module 1

This section identifies the module with a name that characterizes the nature of the module along with describing its type such as, submodule, process, data storage, etc. This section also provides a description of its purpose by identifying the requirements for which it was created. This section should also include a description of requirements that must be met but are not in the requirement specifications.

4.1.1.1 Processing

This section shall describe the algorithm used to perform the task assigned to the module and the associated contingencies. It represents the most detailed level of the functionality of the module. A pseudocode representation can be used to present the processing of the module. The description should include timing, sequences of events, prerequisites for process initiation, priority of events, conditions, looping, and termination criteria. The handling of contingencies should describe the action taken in the case of overflow conditions or in the case of a validation check failure.

4.1.1.2 Interfaces

This section shall identify the interfaces to the module which specifies the requirements for interacting and the methods of interactions. The methods of interaction include the means for invoking or interrupting the module, for communicating through parameters, common data areas or messages and direct access to internal data. The requirements for interacting include the communications protocol, data format, acceptable values, and the meaning of each value.

4.1.1.3 Dependencies

This section describes the relationships of this module with other modules. These relationships are often shown as structured charts, data flow diagrams, and transaction diagrams. The description shall include the timing characteristics, and the conditions for interacting.

4.1.1.4 Resources

This section describes elements used by the module that are external to 'component name' that are needed to perform its function. The mechanism for interacting with these elements is described in this section. This section provides information about physical devices (printers, memory, etc.), software services (libraries, OS services), and processing resources (CPU cycles, buffers, etc.). It should include the usage of resources such as physical sizes of buffers or process time at which resources are acquired. This section should also identify potential race and deadlock conditions as well as management facilities.

4.1.1.5 Data

This section shall provide a detail description of the internal data used by the module. All data formats and the use of the data are presented here. The description should include the initial values, size of the data elements. The data structures that represent the data (i.e. arrays memory partition) should be identified in this section. The use and meaning of the data should be specified. The description should include such things as: static, dynamic, shared, control parameter, loop iteration count, or pointer. The information should also include a description for data validation needed for the module.

4.1.1.6 Submodules

This section shall identify the submodules. A diagram of the module breakdown/architecture should be included here. The remaining portion of this section shall provide the detail information for each module with appropriate pseudocode needed for design reviews and development of 'component name'.

4.1.1.6.1 Submodule 1

4.1.1.6.1.1 Interface Design

4.1.1.6.1.1.1 Inputs

4.1.1.6.1.1.2 Outputs

4.1.1.6.1.2 Algorithm (PDL)

4.1.1.6.1.3 Restrictions/Limitations

4.1.1.6.1.4 Local Data Structures

4.1.1.6.1.5 Performance Issues

4.1.1.6.2 Submodule 2

4.1.2 Module 2