

CMSC 447

Software Design Description (SDD)

Revision 2

1	Scope	3
1.1	Identification	3
1.2	System overview	3
1.3	Document overview	3
2	Referenced documents	3
3	CSCI-wide design decisions	3
4	CSCI architectural design	4
4.1	CSCI components	4
4.2	Concept of execution	5
4.3	Interface design	5
4.3.1	Interface identification and diagrams	6
4.3.2	(Project-unique identifier of interface)	6
5	CSCI detailed design	7
5.1	(Project-unique identifier of a software unit, or designator of a group of software units)	8
6	Requirements traceability	9
7	Notes	9
A.	Appendixes	10

1 Scope

1.1 Identification

This Software Design Description (SDD) will be used to describe the proposed system design for version 1.0.0. of the ROBBR system CSCI. No other releases of this system have been made and no new releases are foreseen as of latest document revision.

1.2 System overview

The purpose of the ROBBR system is to allow a user to determine the best location for a new headquarters for a jewel thief syndicate. The system collects publicly available statistics about locations within the United States, and the user provides criteria for filtering and ranking houses within a user-selected state. The system then generates a list of houses, weighed by their adherence to the user specified criteria, and displays it to the user. The ROBBR system is being developed for the customer who will be the sole sponsor, acquirer, and user. The ROBBR system will run on a bootable USB which may be carried by the customer. The I am Root Software Engineering Team shall be the sole developers of the software, and any reference to the developer will refer to the I am Root Software Engineering Team. For privacy and security reasons, the customer is not identified.

1.3 Document overview

This SDD for the ROBBR system will detail various system interface and architectural details, their purpose, and the communications between them. The document will include graphical aids, which will be used as reference in the construction of the ROBBR system. The document will also include requirements traceability documentation, which will refer to the SRS.

2 Referenced documents

Number	Title	Revision	Date
1	Software Development Plan	2	4/19/2018
2	Software Requirement Specification	3	4/19/2018

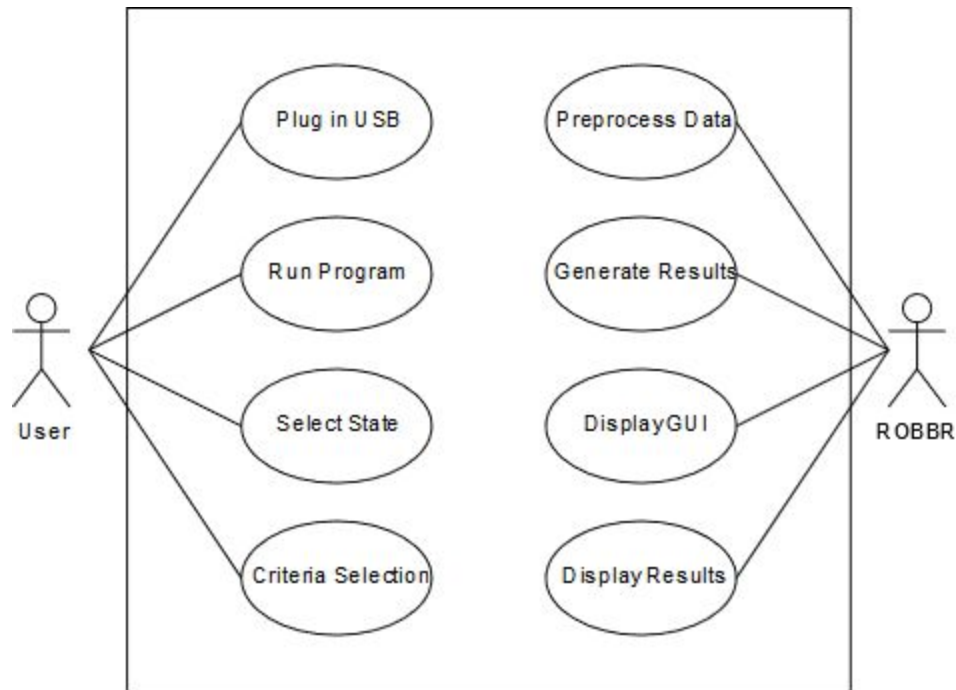
3 CSCI-wide design decisions

This section shall be divided into paragraphs as needed to present CSCI-wide design decisions, that is, decisions about the CSCI's behavioral design (how it will behave, from a user's point of view, in meeting its requirements, ignoring internal implementation) and

other decisions affecting the selection and design of the software units that make up the CSCI. If all such decisions are explicit in the CSCI requirements or are deferred to the design of the CSCI's software units, this section shall so state. Design decisions that respond to requirements designated critical, such as those for safety, security, or privacy, shall be placed in separate subparagraphs. If a design decision depends upon system states or modes, this dependency shall be indicated. Design conventions needed to understand the design shall be presented or referenced. Examples of CSCI-wide design decisions are the following:

- a. Design decisions regarding inputs the CSCI will accept and outputs it will produce, including interfaces with other systems, HWICs, CSCIs, and users (4.3.x of this DID identifies topics to be considered in this description). If part or all of this information is given in Interface Design Descriptions (IDDs), they may be referenced.
 - b. Design decisions on CSCI behavior in response to each input or condition, including actions the CSCI will perform, response times and other performance characteristics, description of physical systems modeled, selected equations/algorithms/rules, and handling of unallowed inputs or conditions.
 - c. Design decisions on how databases/data files will appear to the user (4.3.x of this DID identifies topics to be considered in this description). If part or all of this information is given in Database Design Descriptions (DBDDs), they may be referenced.
 - d. Selected approach to meeting safety, security, and privacy requirements.
 - e. Other CSCI-wide design decisions made in response to requirements, such as selected approach to providing required flexibility, availability, and maintainability.
- a. The system will take input in the form of APIs, which will be stated later, as well as user input. For the input of the desired state a dropdown menu will be used. Crime rate, median household income, and distance from the center of mass of jewelry stores per state will be on a per ZIP code basis and shall be input by the user by a slider, selecting which quintile they want.
 - b. After receiving input in the form of dropdown menus and sliders, the data will be used as input for a weighting algorithm which will then provide a list of ZIP codes within the time constraint of 30 seconds.

3.1 Use Cases



Use Case Name: Plug in USB	Priority: 1. High	Level: Overview
Summary: User will plug in the USB into any computer system, which is a key function of the system since the software is contained on the USB.		
Goal: The successful execution of the ROBBR software.		
Success Measurement: The system is successfully executed once the User inserts the bootable USB 3.0 flash drive into any computer system of the user's choosing.		
Preconditions: <ul style="list-style-type: none">• The User has the bootable USB 3.0 flash drive in their possession.• A computer with a USB port is readily available to the User.• The bootable houses the ROBBR software. Postconditions: <ul style="list-style-type: none">• The software stored in the bootable will run. Trigger: <ul style="list-style-type: none">• The User begins their workflow at this Use Case.		
Primary Actor(s): ROBBR User	Secondary Actor(s): None	

4 CSCI architectural design

This section shall be divided into the following paragraphs to describe the CSCI architectural design. If part or all of the design depends upon system states or modes, this dependency shall be indicated. If design information falls into more than one paragraph, it may be presented once and referenced from the other paragraphs. Design conventions needed to understand the design shall be presented or referenced.

4.1 CSCI components

This paragraph shall:

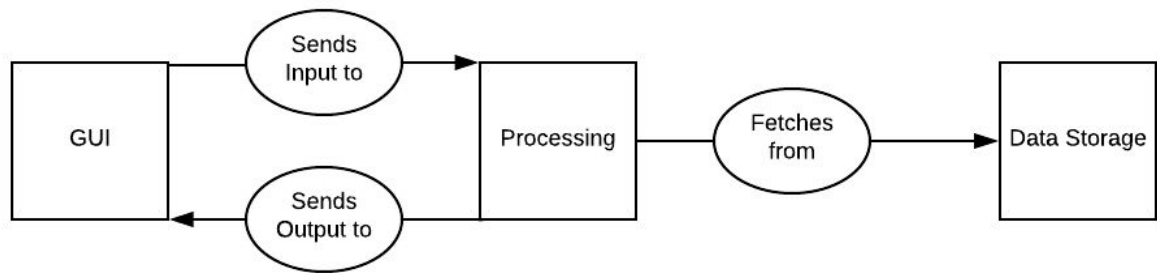
- a. Identify the software units that make up the CSCI. Each software unit shall be assigned a project-unique identifier.

Note: A software unit is an element in the design of a CSCI; for example, a major subdivision of a CSCI, a component of that subdivision, a class, object, module, function, routine, or database. Software units may occur at different levels of a hierarchy and may consist of other software units. Software units in the design may or may not have a one- to-one relationship with the code and data entities (routines, procedures, databases, data files, etc.) that implement them or with the computer files containing those entities. A database may be treated as a CSCI or as a software unit. The SDD may refer to software units by any name(s) consistent with the design methodology being used.

The ROBBR CSCI consists of three independent portions of functionality, and three corresponding CSCs:

- Data Storage: Includes both preprocessed data and Google Maps API which are necessary for zip code evaluation
- Processing: Calculates weighted zip code list from stored data and based on user input
- GUI: Takes search parameters from user and displays results of processing

- b. Show the static (such as "consists of") relationship(s) of the software units. Multiple relationships may be presented, depending on the selected software design methodology (for example, in an object-oriented design, this paragraph may present the class and object structures as well as the module and process architectures of the CSCI).



- c. State the purpose of each software unit and identify the CSCI requirements and CSCI-wide design decisions allocated to it. (Alternatively, the allocation of requirements may be provided in 6.a.)

Software Unit	Description	Requirement(s)
Data Storage	Holds a set of files for stored data to be pushed to the Query object	<p>The system contains stored data on all criteria entered by the user for each zip code</p> <p>The system shall apply an algorithm to the data of each zip code of the selected region compared to data input by the user to rank the zip codes in the selected region</p>
Processing	Calculates weighted zip code list from stored data and based on user input, communicates with Data Storage	<p>The algorithm which returns a weighted zip code list will weight each value with a floating point multiplier in front of it</p> <p>The results of the weighting algorithm shall be displayed within 30 seconds of the last user input being received</p> <p>The system shall apply an algorithm to the data of each zip code of the selected region compared to data input by the user to rank the zip codes in the selected region</p>
GUI	Initializes on the start of the program, takes user input and pushes it to Processing, then displays results	<p>The results of the weighting algorithm shall be displayed</p> <p>The system shall provide a slider for the selection of average crime per zip code</p> <p>The minimum distance from the center of mass of jewelry stores in the state selected compared to the center of each zip code will be 25 miles</p>

		<p>The system shall provide a slider for the selection of maximum distance from the center of mass of jewelry stores in the state selected compared to the center of each zip code</p> <p>The system shall provide a slider for the selection of minimum distance from the center of mass of jewelry stores in the state selected compared to the center of each zip code</p> <p>The system shall provide a slider for the selection of maximum median household income per zip code</p> <p>The system shall provide a slider for the selection of minimum median household income per zip code</p> <p>The system shall provide sliders with choices by quintile. The system shall provide a list of the fifty United States, and shall allow the user to select one state.</p>
--	--	---

Identify each software unit's development status/type (such as new development, existing design or software to be reused as is, existing design or software to be reengineered, software to be developed for reuse, software planned for Build N, etc.) For existing design or software, the description shall provide identifying information, such as name, version, documentation references, library, etc.

- d. Describe the CSCI's (and as applicable, each software unit's) planned utilization of computer hardware resources (such as processor capacity, memory capacity, input/output device capacity, auxiliary storage capacity, and communications/network equipment capacity). The description shall cover all computer hardware resources included in resource utilization requirements for the CSCI, in system-level resource allocations affecting the CSCI, and in resource utilization measurement planning in the Software Development Plan (SDP). If all utilization data for a given computer hardware resource are presented in a single location, such as in one SDD, this paragraph may reference that source. Included for each computer hardware resource shall be:
 - 1) The CSCI requirements or system-level resource allocations being satisfied
 The USB will use the processor and ram of what every computer it is plugged into. The storage will be on the usb that has usb 3.0. There is only 4GB of storage available. This is the maximum allowed for persistence in a bootable USB.
 - 2) The assumptions and conditions on which the utilization data are based (for example, typical usage, worst-case usage, assumption of certain events)
 The system that is being used to boot the thumb drive must be able to be bootable. The user must be able to connect to the internet to use the software.
 - 3) Any special considerations affecting the utilization (such as use of virtual memory, overlays, or multiprocessors or the impacts of operating system overhead, library

software, or other implementation overhead)

Speed may be a concern since USB 2.0 can transfer data at 480MB/sec and USB 3.0 can transfer data at 5GB/sec. If the system the usb is booted on only has 2.0 then speed might be a concern. Insufficient RAM to handle the program may also pose a problem, since the USB probably doesn't support

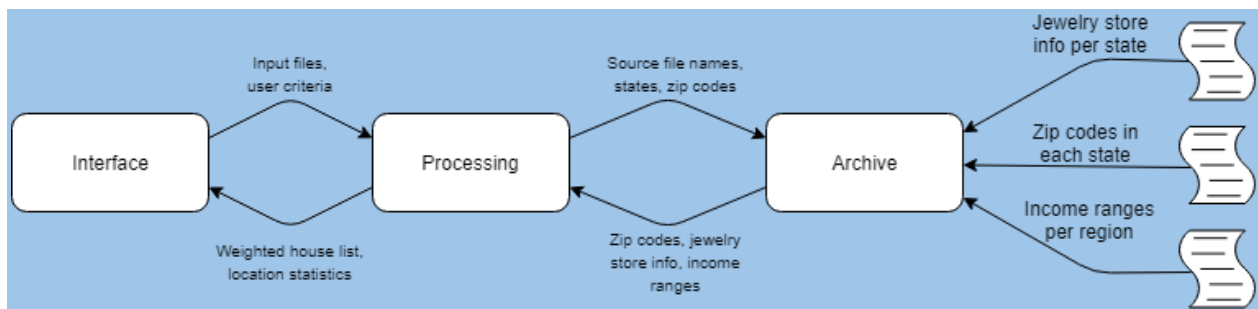
- 4) Identify the program library in which the software that implements each software unit The units of measure used (such as percentage of processor capacity, cycles per second, bytes of memory, kilobytes per second)
- 5) The level(s) at which the estimates or measures will be made (such as software unit, CSCI, or executable program) is to be placed.

Software units:

web interaction tool: pulls necessary data off the internet and processes it into a usable form consists of a downloader that retrieves the data and a processor that formats it as desired.

4.2 Concept of execution

This paragraph shall describe the concept of execution among the software units. It shall include diagrams and descriptions showing the dynamic relationship of the software units, that is, how they will interact during CSCI operation, including, as applicable, flow of execution control, data flow, dynamically controlled sequencing, state transition diagrams, timing diagrams, priorities among units, handling of interrupts, timing/sequencing relationships, exception handling, concurrent execution, dynamic allocation/deallocation, dynamic creation/deletion of objects, processes, tasks, and other aspects of dynamic behavior.



Once the ROBBR program starts the interface will pop up to handle interaction with the user. Upon the user selecting a state the interface will send the state to the processing unit, which sends it to the archive, which returns all the ZIP codes in that state with any other data it has for those ZIP codes(jewelry stores, income). From there the ZIP codes are passed to the downloader which retrieves the house for sale and crime statistics. The user can then refine their parameters for crime, distance from center of mass, and income, which will be passed to the processing component. The processing component then determines which ZIP codes best fit the search results and returns those ZIP codes, along with houses for sale in them, to the interface. The interface then lists ZIP codes hierarchically and houses within them and displays them

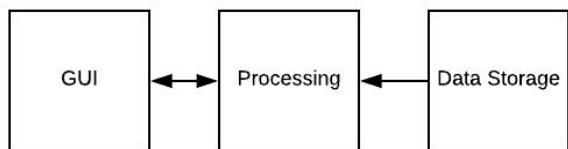
on a map.

4.3 Interface design

This paragraph shall be divided into the following subparagraphs to describe the interface characteristics of the software units. It shall include both interfaces among the software units and their interfaces with external entities such as systems, configuration items, and users. If part or all of this information is contained in Interface Design Descriptions (IDDs), in section 5 of the SDD, or elsewhere, these sources may be referenced.

4.3.1 Interface identification and diagrams

This paragraph shall state the project-unique identifier assigned to each interface and shall identify the interfacing entities (software units, systems, configuration items, users, etc.) by name, number, version, and documentation references, as applicable. The identification shall state which entities have fixed interface characteristics (and therefore impose interface requirements on interfacing entities) and which are being developed or modified (thus having interface requirements imposed on them). One or more interface diagrams shall be provided, as appropriate, to depict the interfaces.



ROBBR has two internal interfaces. The Query interface is represented by the Processing to Data Storage link. The user interface is represented by the Processing to GUI link.

4.3.2 User Interface and Query

This paragraph (beginning with 4.3.2) shall identify an interface by project-unique identifier, shall briefly identify the interfacing entities, and shall be divided into subparagraphs as needed to describe the interface characteristics of one or both of the interfacing entities. If a given interfacing entity is not covered by this SDD (for example, an external system) but its interface characteristics need to be mentioned to describe interfacing entities that are, these characteristics shall be stated as assumptions or as "When [the entity not covered] does this, [the entity that is covered] will" This paragraph may reference other documents (such as data dictionaries, standards for protocols, and standards for user interfaces) in place of stating the information here. The design description shall include the following, as applicable, presented in any order suited to the information to be provided, and shall note any differences in these characteristics from the point of view of the interfacing entities (such as different expectations about the size, frequency, or other characteristics of data elements):

- a. Priority assigned to the interface by the interfacing entity(ies)
 - 1) Data Storage -> Processing
 - 2) GUI <-> Processing
- b. Type of interface (such as real-time data transfer, storage-and-retrieval of data, etc.) to be implemented
 - 1) The Processor to Data Storage interface is a Data Fetch interface.
 - 2) The GUI to Processing is a real time data transfer which passes the search criteria to Processing to be handled as well as returning the search results back to the GUI to be displayed
- c. Characteristics of individual data elements that the interfacing entity(ies) will provide, store, send, access, receive, etc., such as:
 - 1) Data Storage -> Processing
 - a) From Data Storage processing receives a set of CSV with zip code sorted data.
 - 2) GUI <-> Processing
 - a) From GUI Processing receives search criteria including
 - A state name(string)
 - A min value for household income (int)
 - A max value for household income (int)
 - A target crime rate quintile (int 1-5)
 - A minimum distance from the Center of Mass of jewelry stores in the state
 - A maximum distance from the Center of Mass of jewelry stores in the state
 - 3) Names/identifiers
 - a) Project-unique identifier
 - b) Non-technical (natural-language) name
 - c) DoD standard data element name
 - d) Technical name (e.g., variable or field name in code or database)
 - e) Abbreviation or synonymous names
 - 4) Data type (alphanumeric, integer, etc.)
 - 5) Size and format (such as length and punctuation of a character string)
 - 6) Units of measurement (such as meters, dollars, nanoseconds)
 - 7) Range or enumeration of possible values (such as 0-99)
 - 8) Accuracy (how correct) and precision (number of significant digits)
 - 9) Priority, timing, frequency, volume, sequencing, and other constraints, such as whether the data element may be updated and whether business rules apply
 - 10) Security and privacy constraints
 - 11) Sources (setting/sending entities) and recipients (using/receiving entities)
- d. Characteristics of data element assemblies (records, messages, files, arrays, displays, reports, etc.) that the interfacing entity(ies) will provide, store, send, access, receive, etc., such as:

- 1) Names/identifiers
 - a) Project-unique identifier
 - b) Non-technical (natural language) name
 - c) Technical name (e.g., record or data structure name in code or database)
 - d) Abbreviations or synonymous names
 - 2) Data elements in the assembly and their structure (number, order, grouping)
 - 3) Medium (such as disk) and structure of data elements/assemblies on the medium
 - 4) Visual and auditory characteristics of displays and other outputs (such as colors, layouts, fonts, icons and other display elements, beeps, lights)
 - 5) Relationships among assemblies, such as sorting/access characteristics
 - 6) Priority, timing, frequency, volume, sequencing, and other constraints, such as whether the assembly may be updated and whether business rules apply
 - 7) Security and privacy constraints
 - 8) Sources (setting/sending entities) and recipients (using/receiving entities)
- The CSV's from Data storage consist of a list of houses for sale and their price.

5 Requirements traceability

This section shall contain:

- a. Traceability from each software unit identified in this SDD to the CSCI requirements allocated to it. (Alternatively, this traceability may be provided in 4.1.)

Component	Requirement Traceability	SRS Section
GUI	(1) Select Region	3.2.1
GUI	(2) Switch to Filter Criteria Input Mode	3.2.1
GUI	(1) Switch to Select Region Mode	3.2.2
GUI	(2) Input Criteria Selection Sliders	3.2.2
GUI	(2.1) Minimum Household Income Slider	3.2.2
GUI	(2.2) Maximum Household Income Slider	3.2.2
GUI	(2.3) Minimum Distance Slider	3.2.2
GUI	(2.4) Maximum Distance Slider	3.2.2
GUI	(2.5) Average Crime Slider	3.2.2
GUI	(2.6) House Quantity Slider	3.2.2

GUI	(3.0) Criteria Weighting Fields	3.2.2
GUI	(4.0) Switch to Results Output Mode	3.2.2
GUI	(1) Switch to Filter Criteria Input Mode	3.2.3
Processing	(2) Generate Results	3.2.3
Processing	(2.1) Generate Results Quickly	3.2.3
GUI	(3) Display Results	3.2.3
GUI	(3.1) Display Zip Codes as List	3.2.3
GUI	(3.2) Display Houses as List	3.2.3
GUI	(3.3) Display a Map	3.2.3
GUI	(3.3.1) Display Houses on Map	3.2.3
GUI	(3.3.2) Display Region on Map	3.2.3
GUI	(3.3.3) Display Center of Mass on Map	3.2.3
Processing	(1) Results Generation Algorithm Zip Code Statistics	3.3.2
Processing	(2) Results Generation Algorithm Zip Code House Information	3.3.2
GUI	(1) Provide Graphical User Interface	3.3.3
Data Storage	(1) Statistics Format Within File Structure	3.4
Data Storage	(2) House Format Within File Structure	3.4
Environment	(1) Flash Drive System	3.5
Environment	(2) No Logging	3.5
Environment	(1) Computer Guest Operating System Boot Capability	3.6
Environment	(2) Computer USB 3.0 Port Availability for Guest OS Flash Drive	3.6
Environment	(3) Computer Internet Connection	3.6

All requirements apply to at most one component.

6 Notes

This section shall contain any general information that aids in understanding this document (e.g., background information, glossary, rationale). This section shall include an alphabetical listing of all acronyms, abbreviations, and their meanings as used in this document and a list of any terms and definitions needed to understand this document.

A. Appendixes

Appendixes may be used to provide information published separately for convenience in document maintenance (e.g., charts, classified data). As applicable, each appendix shall be referenced in the main body of the document where the data would normally have been provided. Appendixes may be bound as separate documents for ease in handling. Appendixes shall be lettered alphabetically (A, B, etc.).

DESCRIPTION/PURPOSE

The Software Design Description (SDD) describes the design of a Computer Software Configuration Item (CSCI). It describes the CSCI-wide design decisions, the CSCI architectural design, and the detailed design needed to implement the software. The SDD may be supplemented by Interface Design Descriptions (IDDs) and Database Design Descriptions (DBDDs).

APPLICATION/INTERRELATIONSHIP

Portions of this plan may be bound separately if this approach enhances their usability. Examples include plans for software configuration management and software quality assurance.

The Contract Data Requirements List (CDRL) should specify whether deliverable data are to be delivered on paper or electronic media; are to be in a given electronic form (such as ASCII, CALS, or compatible with a specified word processor or other support software); may be delivered in developer format rather than in the format specified herein; and may reside in a computer-aided software engineering (CASE) or other automated tool rather than in the form of a traditional document.

PREPARATION INSTRUCTIONS

General instructions.

- a. Automated techniques. Use of automated techniques is encouraged. The term "document" in this means a collection of data regardless of its medium.
- b. Alternate presentation styles. Diagrams, tables, matrices, and other presentation styles are acceptable substitutes for text when data required can be made more readable using these styles.
- c. Title page or identifier. The document shall include a title page containing, as applicable: document number; volume number; version/revision indicator; security markings or other restrictions on the handling of the document; date; document title; name, abbreviation, and any other identifier for the system, subsystem, or item to which the document applies; contract number; CDRL item number; organization for which the document has been prepared; name and address of the preparing organization; and distribution statement. For data in a database or other alternative form, this information shall be included on external and internal labels or by equivalent identification methods.
- d. Table of contents. The document shall contain a table of contents providing the number, title, and page number of each titled paragraph, figure, table, and appendix. For data in a database or other alternative form, this information shall consist of an internal or external table of contents containing pointers to, or instructions for accessing, each paragraph, figure, table, and appendix or their equivalents.

- e. Page numbering/labeling. Each page shall contain a unique page number and display the document number, including version, volume, and date, as applicable. For data in a database or other alternative form, files, screens, or other entities shall be assigned names or numbers in such a way that desired data can be indexed and accessed.
- f. Response to tailoring instructions. If a paragraph is tailored out of this document, the resulting document shall contain the corresponding paragraph number and title, followed by "This paragraph has been tailored out." For data in a database or other alternative form, this representation need occur only in the table of contents or equivalent.
- g. Multiple paragraphs and subparagraphs. Any section, paragraph, or subparagraph in this DID may be written as multiple paragraphs or subparagraphs to enhance readability.
- h. Standard data descriptions. If a data description required by this document has been published in a standard data element dictionary specified in the contract, reference to an entry in that dictionary is preferred over including the description itself.
- i. Substitution of existing documents. Commercial or other existing documents, including other project plans, may be substituted for all or part of the document if they contain the required data.