Moving From Users, Through Use Cases To Requirements

Lynne Davis, Tim DiLauro, Mark Evans,
Siri Jodha Singh Khalsa, Ruth Duerr, Anne Thessen

A Data Conservancy White Paper

Version History

Version Number	Date	Editor	Comment
0.5	10/6/10	Lynne Davis	Join of Tim's additions to front matter, Mark's addition of Appendix A and B, and Lynne's diagram updates and examples, and general content edits.
0.6	10/8/10	Lynne Davis	Semi-final draft. Structural changes to the document for readability. Executive Summary added
0.7	10/11/10	Lynne Davis	Clarifying points in 5.1, from TD, MM, KK
0.8	10/11/10	Lynne Davis	Final edits and TOC

Overview

The Data Conservancy (DC) is developing an organizational and technical framework of systems and services that protect and provide long-term access to scientific data and related content. The purpose of this document is to describe the process being followed by those IS/CS researchers using the User-Centered Design (UCD) approach for defining DC Service (DCS) requirements. It is intended for use by those researchers and the IRD team as a reference to coordinate efforts in this highly iterative process.

A core part of the Data Conservancy's framework is the Data Conservancy Service (DCS), the technical infrastructure that will underpin it. To ensure the utility and usability of this system, DC is developing a requirements management (RM) process, which both informs and is informed by the user studies component of the project. The RM process, in conjunction with an iterative user-centered approach to design and evaluation, will document and assure that user needs are met. In addition, this RM approach must support the concept of traceability, as articulated in PMP 3.0 Section 3.2.

Viewed from a high level, the process involves the following cyclical and iterative activities:

- Capturing available work products and research outputs;
- Reviewing these and, where possible, extracting high-level use cases and requirements;
- Creating refined use cases and requirements using the guidelines in Appendix A and Appendix B;
- Designing for selected use cases and requirements;
- Creating testable mockups, prototypes, pilot versions at iterative review points
- Evaluating testable components with user base
- Adjusting use cases and requirements
- Unifying similar use cases and requirements;
- Reviewing and prioritizing unified use cases and requirements for next release(s);
- Developing DCS based on design.

One of the strengths of this effort is the concurrent work on both the development and research fronts. As such, we anticipate that the research effort will inform the development effort as appropriate and as research findings become available.

This document describes the UCD-based process for gathering sources of information about users, transitioning them toward progressively more highly structured forms using the Use Case methodology, and stating them as formal requirements. The use cases and requirements inform design and guide DCS development and evaluation.

Information about users and what they do are captured as, e.g., personas, stories, scenarios, profiles and case studies. A use case is a methodology used in system analysis to identify, clarify, and organize user and system requirements. Use cases also house metrics to assure ongoing alignment with user needs. Use cases are an important source of user requirements. A requirement may serve a user directly or indirectly via system execution. A user requirement is a structured expression of a user need, is expressed from the user's perspective, and meets established criteria. (See also Appendix D.)

2.0 Introduction

Purpose

As stated above, the purpose of this document is to describe the process being followed by those IS/CS researchers using the User-Centered Design (UCD) approach for defining DC Service (DCS) requirements. It is intended for use by those researchers and the IRD team as a reference to coordinate efforts in this highly iterative process.

Scope

The initial scope of this document will encompass the role that the user-centered design (UCD) approach being led by NCAR plays in the RM process. This includes input from the Science Working Groups.

While this version does not currently describe the roles of the UCLA and Illinois teams in the RM process, it is our intention to do so eventually (i.e., shortly after the October 27th meeting) so that we have a single document describing our entire RM process.

This document will not describe how the Infrastructure Research and Development (IRD) Team will develop software, but will discuss the IRD Team's interactions with others involved in the requirements analysis process. Though traceability will be supported by the overall RM process, it will not be a focus of this document.

3.0 Definitions

A common vocabulary and shared understanding is essential for guiding disparate teams in a common process. To that end, Appendix D provides definitions of terms used in this document. Terms defined include Stories, Case Studies, Profile, Scenario, Persona, Use Case (and various levels and distinctions), Initial Analysis, Formative Evaluation Tasks, Requirements (User, Science, and System).

4.0 Sources of Information

Entry-level source content may take a variety of forms. Source content may be broad or narrow in scope, context-rich or sparse and may be domain-specific or general. It should present at least one desired or necessary user activity for DC to support. These needs are fashioned into requirements through iterative use case refinement and stages of evaluation. The best sources of information come from first-hand accounts, but useful information also comes from less direct accounts. It should be made clear how the information was obtained, as this can play a role in prioritization. In a rough order, most to least direct:

Self; Quotes; Transcripts; Observations of a subject at work; Interviews you have conducted; Focus Groups and Workshops you have attended; Expert Accounts, e.g. of individuals in the lab or memorable experiences; Case Studies and Research Publications

Source information should provide a description of the immediate situation about which it is being written (a story, scenario or summary) and a description of the person or persons doing the work, e.g.: their goals, work community, type of science being done, motivations, career stage, job description, work habits, work preferences, etc. Source information should include specific information as examples to inform the support structures needed in the DC to accomplish their goals: e.g., what they are doing, where, when, how, why? How is data involved? What is the problem or challenge for them personally? Precise descriptions of their use of data, tools involved, and data workflows are particularly useful. These accounts can be uploaded to the Source Documents area of the DC requirements management tool (Contour) or uploaded into a science domain directory within Personae, Scenarios, or High Level Use Cases (not system). Contour provides the means to link and trace sources.

Sets of legacy requirements for functionality that are established, mature, well-accepted, and well-documented in support of user needs and are currently used in similar systems may be suitable for review and ingest straight to User or System requirements. These, however, should be linked to at least one use case for evaluation purposes.

5.0 Process

5.1 Overview

Requirements Generation Process

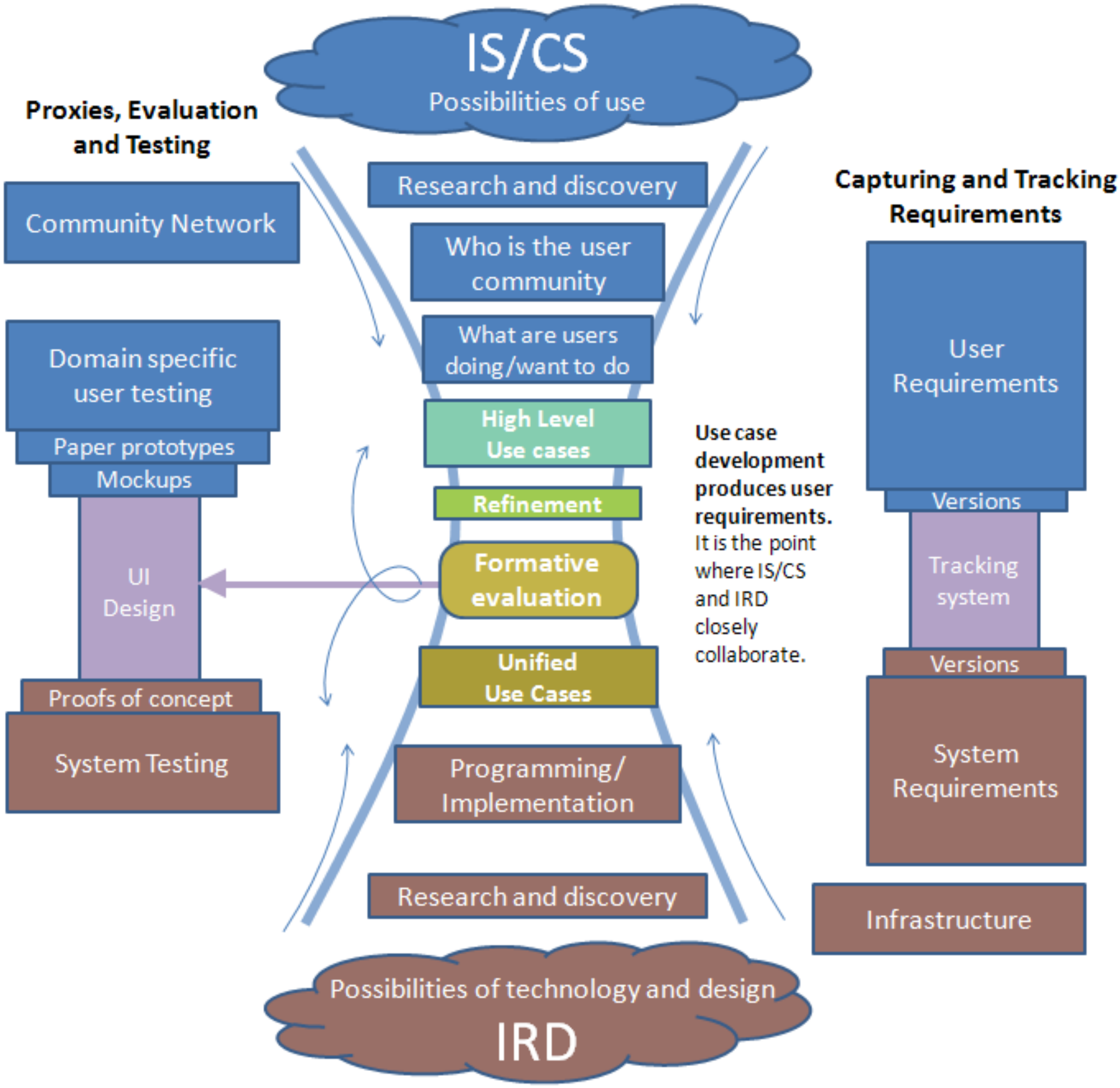


Figure 5.1 Over-arching view of how requirements are generated and implemented
User requirements can be expressed at increasingly more detailed levels – which correspond to the levels of use case development. The richest user context exists at top levels (blue), while the richest system context exists in the bottom (brown) levels.

Figure 5.1 provides an over-arching view of some IS/CS (blue) and IRD (brown) focal areas, the point at which those areas merge to exchange information, and indicators of design/development checkpoints and iteration. It outlines a very high-level cyclical process workflow that accommodates both a user-centered approach to determining requirements and design and the Unified Modeling Language (UML) structure adopted as part of the requirements management process.

In this model, parallel work can take place within the IRD (brown) and IS/CS (blue) camps with checkpoints at intervals. The overlapping, "handoff" point between community /user-centered research and DC product development happens at the use cases, depicted in the central region of the diagram. It is at this juncture between IRD and IS/CS activities where requirements are formalized, design takes place, and evaluation is conducted.

The upper (blue) and central (multicolored) vertical region depicts the process of transforming IS/CS and science working groups work products and research findings into requirements. It is important to note that this activity does not, in and of itself, impose particular deadlines for the creation of these research outputs; it is simply a point at which available outputs can be reviewed and brought into the RM process. New RM artifacts may be extracted from the documents in the scenario or persona areas and take the form of high-level use cases. These can be made increasingly more explicit through the use case refinement process (described elsewhere in this document). It also illustrates where in the process to have a plan for formal, specific checkpoints or reviews with the user community to assure functional alignment with their needs (formative evaluation). And equally importantly, it is the point at which IS/CS and IRD begin to work together to express accurately and completely the fine-grained requirements, priorities and release features so that IRD can confidently assume the responsibility for expressing and tracking them as development tasks for each stage of release.

5.2 Proxies, Evaluation and Testing

Visual proxies, such as models, workflows, paper prototypes, user interface mockups, as well as pilot builds, functional prototypes all assist in being able to conduct user testing at iterative stages of development. Proxy creation, evaluation and testing components are shown along the left side of Figure 5.1, and relate most directly with the Formative Evaluation part of the overall process. User testing and system testing are conducted at scheduled intervals during the design & development cycles to allow opportunity for introducing or adjusting requirements based on ongoing research findings. Formative Evaluations assure continued alignment between system and user requirements by engaging the user communities. These stages make use of increasingly higher fidelity, tangible representations of a design to assess the use of the system and its functionality.

Results of evaluation give evidence for modifying use cases that drive the technology and design of the system.

5.3 Capturing and Tracking Requirements

Along the right side of Figure 5.1 are components involved in capturing and tracking requirements and product releases. User requirements can come from documented sources of all research activities which study the proposed user communities. They state what the user requires to preserve, support or advance their work, not how the DC will deliver it. Tracking between sources, use cases, evaluation metrics, and requirements is made possible by the adopted Requirements Management tool, Contour.

5.4 Use Case Refinement Process

Figure 5.2 outlines a process of turning Scientific User/Community research findings, scenarios and stories into testable sets of structured use cases and measureable tasks.

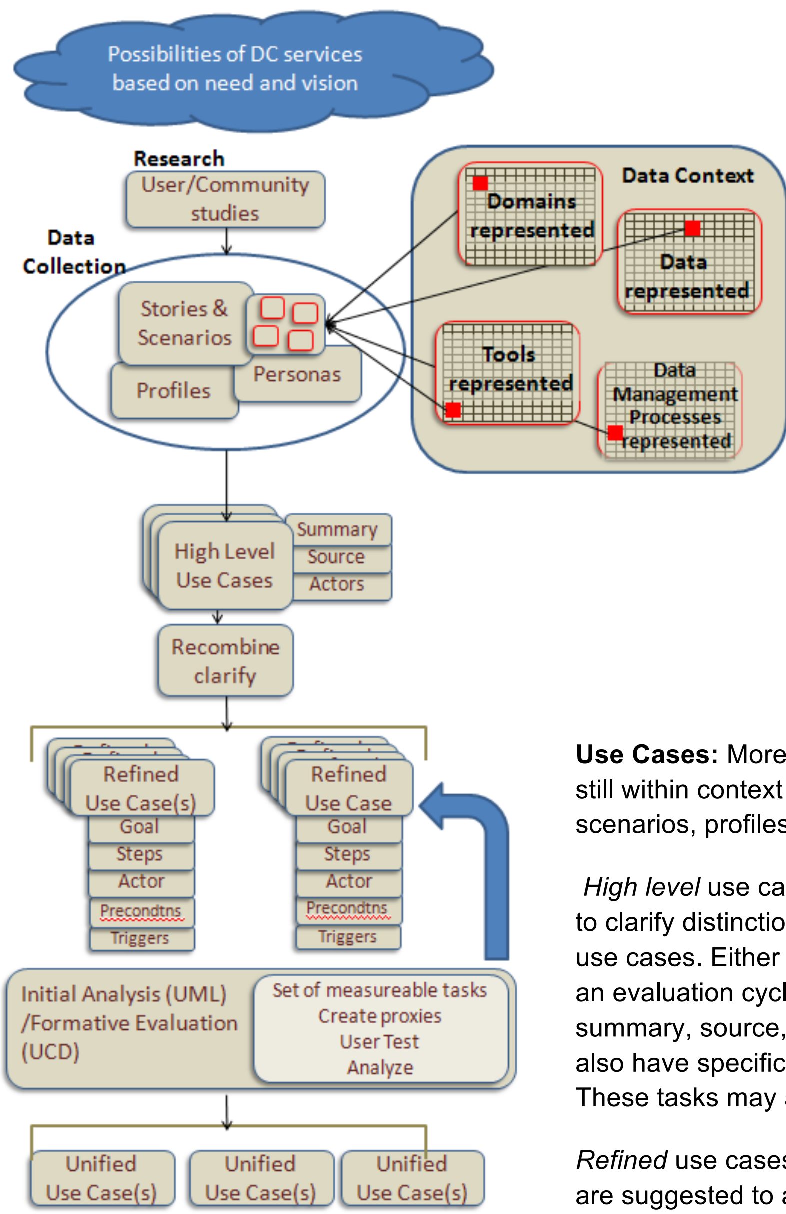


Figure 5.2
Workflow of Use Case Refinement Process

Possibilities: The universe of how DC might serve all Sciences Research Practice

Research: Interview, Observation, Workshop –what are the most important aspects

Data Collection: Information on who are the people; how they work with data; their practices and tools; characteristics of data used and use requirements over time. Information is structured as profiles, representative personas and scenarios.

Data Context: Aspects of data and data use that might be represented in the source material: Science Domains, Data Types, Tools used, Data Management Processes.

Use Cases: More specific instances of how data is used, still within context of the domain and based on the scenarios, profiles, personas.

High level use cases are examined for similar cases and to clarify distinctions. These become a new set of refined use cases. Either can be scrutinized by the community in an evaluation cycle. Each high-level use case has a summary, source, and actor (persona in a role) and may also have specific testable tasks for formative evaluation. These tasks may also be stated as User Requirements.

Refined use cases have a goal, descriptions of what steps are suggested to accomplish the goal (not telling how), the actor involved, preconditions assumed and what triggers the activity described by the use case. Refinement may be done within a science domain.

Unified use cases result from an analysis of refined use cases across science domains. They comprise the set of use cases against which a given release of system development will be done. These can change over time at specific intervals, as requirements that feed into them change.

5.5 Initial Analysis/Formative Evaluation Tasks

The stages of Initial Analysis/Formative Evaluation are scheduled review points to assure that use cases continue to represent the totality of the conclusions captured by IS/CS research studies. Through levels of refinement, they becoming increasingly more explicit in terms of goals, steps to accomplish, and other parameters needed for identifying requirements.

Evaluation and iteration are what assure opportunity for inclusion of new-found needs. As new or changed user needs are discovered and articulated via ongoing research efforts in the IS/CS, they will be processed into new or updated use cases, measureable tasks and user requirements.

Vital to meaningful evaluation is the preservation of the original contexts of use. The requirements management tool preserves the context of use via links between components, e.g., use cases, personas, scenarios and requirements. These links point to specific contexts and are appropriate for creating representative tasks for evaluation. Contour is the RM tool selected to manage this process. Refined use cases can require entry of Initial Analysis statements and/or tasks for evaluation. User requirements can also be repurposed for evaluation.

5.6 Design

The interdependence between the use case and its metric is where system requirements begin to support user requirements and where design really begins. Once science use cases have been identified, organized, and traced to their source context, they are documented. User interface affordances can then be designed. Early proxies of system functionality may be represented as, e.g., steps, lists, paper prototypes and mockups (i.e., proxies in the figure), and used as aids in user testing.

Task performance during user testing can inform design decision points and critical paths. Placed at development milestone check-points, user testing can help to assess the state of product effectiveness and readiness for release.

6.0 Next actions

This process will be tested and amended as needed based on experiences gained through the pilot being built for the Glacier Photos, the experiences to date in writing use cases and requirements for DCS and experiences anticipated as new requirements come to light during iteration.

7.0 Examples

The examples here illustrate the workflow from source materials to use cases and requirements.

7.1 Life Science

Personal Profile – Interview: S-1

S-1 is a mid-career oceanographer at Woods Hole Oceanographic Institution. She is funded by NOAA for 3 years to investigate the effects of climate change on marine food webs. To do so, she goes to sea several times per year to collect field data to compare with historical data and identify trends. She especially enjoys these trips and the preparation involved but does not enjoy mundane tasks of data entry which she gladly assigns to her hand-picked support team. Her team includes a science diver who also acts as a data manager and who deals with routine activities with the many instruments and the volumes of data they produce. S-1 is looking for ways to streamline her work so she can focus on discovery.

She aspires to contribute in significant ways to the understanding of impacts that changes in environment have on marine life. She is thorough and precise in her work and very organized in both her thought processes and methods. She enjoys being recognized by her peers for her significant contributions and as an authority in her field, and is keenly aware of the culture of competition for publication and to gain citations to advance in her career. It is important to her to be the guardian of her data and its use.

In her spare time she is often a lead actor in local theatre productions and enjoys outdoor activities. When she takes business trips, she likes to stay with friends whenever possible.

Data Types

- Observational
- Images
- Species occurrence
- Molecular

Data Tools

- WHOI provides some data support
- Federal Geographic Data Committee (FGDC) metadata guidelines
- MERMAid tool for metadata management
- NODC as repository for uploaded data

Data Sharing

- Uploads to NODC as required by funder
- Uploads to WHOI data store
- Publication
- Will send out data in spreadsheets if asked and if she has the time to put them together

SUMMARY

S-1 has been funded by NOAA to do a series of cruises in the Gulf of Mexico to investigate the effect of climate change on marine food webs. She collects field data about the biology and chemistry of the water. She would like to compare her field data with historical data to identify trends. During her cruises, she collects data on temperature, salinity, irradiance and fluorescence using instruments on board. Her research team collects water samples for later analysis in the laboratory (Nitrate, nitrite, phosphate, ammonium, silicate and plankton counts, molecular). Some of these data are entered directly into a spreadsheet, but some are recorded onto printed data sheets. Much of her data comes off of a machine and must be transformed to be useful. Each instrument and analysis has its own limits of detection and precision. Laura's lab uses the Federal Geographic Data Committee (FGDC) metadata guidelines and the MERMAid tool for metadata management. NOAA requires that she upload her data into the NODC, but she has never downloaded data from NODC because of poor usability. She would like to be able to go to one place, download all phosphate measurements made in the Gulf of Mexico and receive that data in a file, formatted to her specifications. Laura would share her data in the DCS as long as she could keep track of usage and gain citations.

PERSONA – LAURA (This would be compiled from similarities in habits, motivations, preferences, life experience, desires, and goals of more than one person. …. May or may not be science domain-specific.)

SCENARIO – The Laura persona may encompass many scenarios and produce numerous requirements. Personas could also lead directly to use cases and/or requirements.

USE CASE – High Level

<table>
<tr>
<td>Name: List of organisms from query
ID: UC-10

Domain: Life Sciences
Author: Anne Thessen
Source: Laura, SCEN-3

Use Case Summary:
Laura wants a list of organisms that were observed in the northern Gulf of Mexico during years with higher than average precipitation in the Mississippi River watershed.

Initial Analysis: Laura can discover and download a list of organisms that meet the following criteria that she specifies in a query to DC:
1. Organisms were observed in the northern Gulf of Mexico, AND
2. Organisms were observed during years when the Mississippi River watershed experienced higher than average precipitation</td>
<td rowspan="2">High level user requirements suggested:

1. Data query (ID: RQ-24)
Description: Users should be able to query the system through an interface that allows a user to request a specific parameter (the parameter may have several names) collected from a specific place during a specific time.

2. Data query return (ID: RQ-25)
Description: The system shall return to a user the results of their query in a form they can see online first and then download in a format of their specification. The user should be able to specify what order the columns and rows should be in before download. Users should be able to specify the units they want each parameter in.</td>
</tr>
<tr>
<td>Name: Using data in a lesson plan
ID: UC-51

Domain: Life Science
Author: Anne Thessen
Source: Celia

Use Case Summary: Celia wants to find data on dates when bud burst occurred over the last 10 years in areas around the world. She wants to find</td>
</tr>
</table>

similar data to what she has collected in her area over the same period, aggregate the two sets for a worksheet she is preparing for her 6th grade classroom so her class can compare trends. Initial Analysis: Celia can discover a dataset that contains a list of dates when bud burst occurred across the world, extending over the previous 10 years time.	

USE CASE - Refined

Goal: Enter a query for the set of organisms that correspond to a defined environmental feature

Actor: data requestor

Pre Conditions:

1. Data about precipitation in the Mississippi river water shed is accessible to the system
2. Species observation data are available to the system from the Gulf of Mexico
3. User knows average precip in MS riv watershed

Triggers:

User tells system she wants to initiate a query

Normal Path: (Variation paths can also be provided.)

1. User tells system the parameter she wants: a list of taxa
2. User specifies geographic range (default is all data)
3. User specifies temporal range (default is all data)
4. User applies filter based on value of another parameter (precipitation)
5. System notifies user if data are available
6. System gives user a preview of data
7. User downloads data

Initial Analysis/Formative Evaluation

Ref: Laura, UC-10, Scen-3; Celia, UC-51

USE CASE - Unified

This is much like a Refined use case in structure, but represents refined use cases across domains when possible.

7.2 Social Science

Interview – S2 and S3

A team of researchers working on a meta-analysis of urban vulnerability case studies would like to be able to contact other social scientists online to discuss their published work and gain access to studies they are doing in the field of urbanization and climate change research. They would like for DC to provide a network for this purpose. The core of this network is a library system which collects journal articles, conference and working papers, and relevant data sets in this area. This could provide a very useful tool for their future research. In addition, this intellectual community could also promote collaborations among researchers from different disciplines and regions.

7.3 Earth Science

NSIDC's Glacier Photo Database system will be extended so that on an order placed for glacier photographs held at JHU, the system will automatically retrieve those images and add them to the rest of the users order. This change should be transparent to end users.

Uses legacy search systems with new connections to external databases.

Appendix A: Use Case Template

The following template is to be used within the Contour tool for managing use cases.

Field	Description	Mandatory
Name	A unique name given to the use case	Yes
ID	A Contour generated identifier. Will have the prefix "UC"	
Type	The type of use case that is being described. Available choices are "Science" and "System"	Yes
Domain	The scientific domain that the Use case is based on. Available choices are "Life Sciences", "Earth Sciences", "Social Sciences", "Astronomy", "Cross Domain" and "N/A". The list of available options may increase as more domains are encompassed by DC	Yes
System Release	The system release of the DC software that is providing functionality to support the use case.	
Status	Captures the status of the use case. E.g. "New" , "Draft", "Completed" etc	
Author	The DC team member who created the use case	Yes
Source	The source of information used to develop the use case. This could be a set of links to other use cases, source documents or other information on the WIKI	
Goal	The intended objective of the use case	Yes
Actors	The main participants in the use case. These can be classified into primary actors and secondary actor. A primary actor is directly affected by the use case, and a secondary actor is indirectly affected. The actors should be specified in terms of system roles once the use case has been fully refined.	
Use Case Summary	A descriptive narrative of the use case. When a use case is first specified as a high level use case, most of the detail will be captured in this field	Yes
Pre Conditions	Any conditions, assumptions or system state that has to exist before the use case can be executed	
Triggers	The event or action that causes the use case to be executed.	
Normal Path	The set of steps that should be followed in order to successfully execute the use case.	
Alternate Paths	If the use case contains branching or exceptions then these should be captured as a set of alternate steps.	
Post Conditions	And conditions, or state that exist once the use case has been executed	
Notes	Any additional notes or supplementary information that the author wished to specify.	
Initial Analysis	To be used when moving from a high level use case to refined use cases. This field is required for provenance and traceability	
Created Date	The date the use case was first created. This field is populated by Contour	Yes

It is not necessary to populate all of the fields during the capture of a high level use, however it is expected that all fields are populated once refinement has been completed.

Appendix B: What makes a good requirement

Overview

A good set of requirements is fundamental to the success of any software engineering project. Many projects have resulted in failure because the requirements have been poorly specified and, as a result, a system is built to an incorrect specification. This appendix will address what makes a good requirement.

At a high level, a good requirement is something that is ***necessary*** (i.e. a need has been identified or is required to meet the overall objectives), ***verifiable, attainable*** and ***clear and concise***. In order to be verifiable the requirement must state something that can be demonstrated by examination (inspection), test, analysis, or demonstration. To be attainable a requirement must be technically feasible and fit within the program constraints. In order to be clear and concise a requirement must have the following attributes:

- only contain a single requirement (No use of "all" , "and", "any", "or")
- be of limited length (30-40 words MAX)
- must be easy to read and understand
- must be unambiguous and not open to multiple interpretations
- must not contain definitions, or reasons for its need

There are several other criteria that a good requirement must meet.

Complete – Contains all the information that is needed to define the need. May include measurement units

Consistent – Does not conflict with other requirements in the requirements specification or create redundancy.

Traceable – Has a unique identity, and can be linked to other artifacts such as use cases, objectives and architectural components.

Implementation free – Defines **what** functions are required and NOT **how** a function should be implemented.

Use of terms

The following terms are used in good requirements

"Shall" used to define a requirement – "*A user **shall** be able to use a password as a means of authentication*"

“Should” used to define goals within a requirement – “*A user **should** get a list of search results within 2 seconds”*

“Will” used to make a statement of fact – “*A user session **will** be terminated after 30 minutes of inactivity”*

In the main the term “Shall” is to be used for requirements

Avoid using ambiguous terms such as:

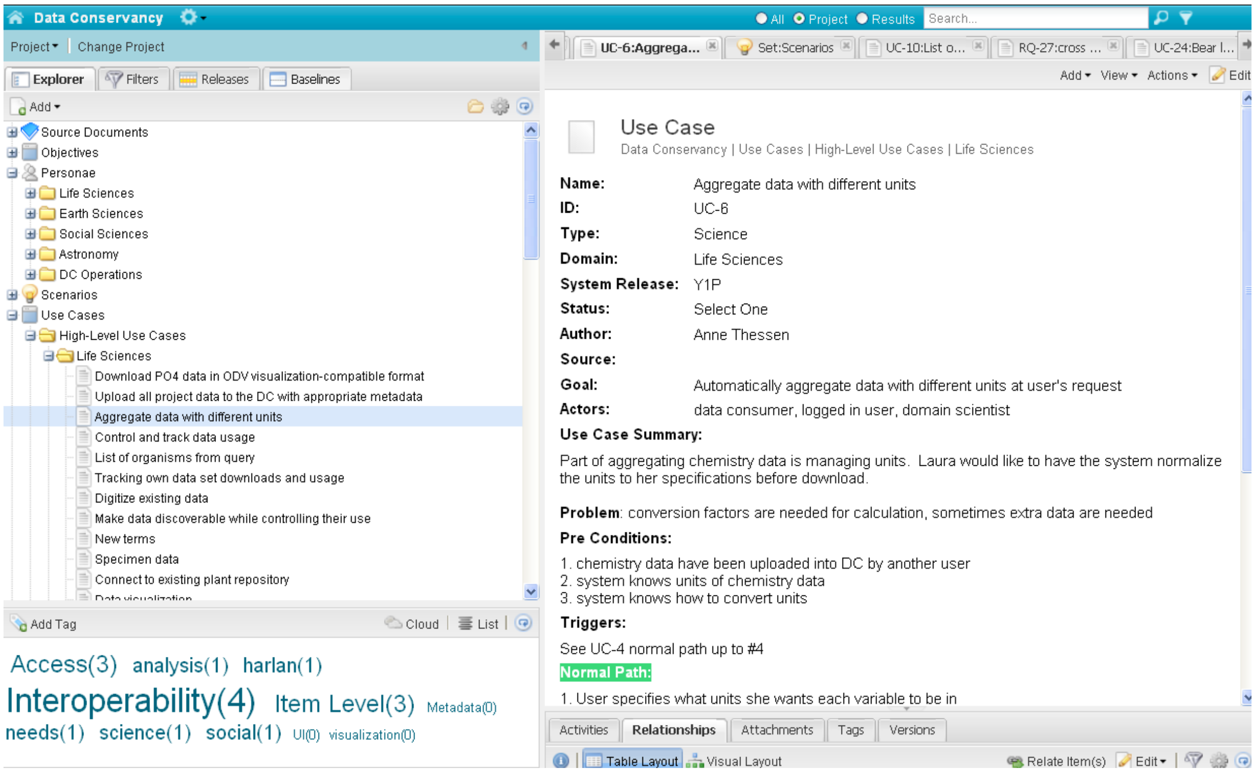
- And /Or
- Not limited to
- Etc
- User friendly
- Easy
- Rapid
- Sufficient
- Adequate

Examples of good and bad requirements

Bad	**Problem**	**Good**
The system must be user friendly.	Not verifiable: How do you measure “user friendly”?	The user interface shall provide menus, help screens , radio buttons, list boxes for user input
All user forms must appear on the monitor quickly.	Not clear: How long is quickly	When a user requests a form, it must appear on the monitor within 1 second.
A user must be able to navigate through a document from start to finish.	Not complete: What is the size of the navigation step?	A user must be able to navigate through a document one page at a time
A user must be able to enter a Dublin core title. A user must be able to enter a subject.	Not consistent: Does the subject need to be Dublin core too?	A user must be able to enter a Dublin core title. A user must be able to enter a Dublin core subject.
A user must be able to see search results instantly.	No Feasible: There will always be some processing time	A user must be able to see a search result within 2 seconds of a request
All users must have 2 factor credentials	Not Necessary: Public users will not need multifactor credentials	All administrative users of the system must have 2 factor credentials
When a user deposits content a java workflow must run to check the manifest	Specifies implementation: It is not necessary to state how.	When a user deposits content the system much perform a check on the manifest.

Appendix C: Contour structure

The following structure exists within Contour.



Appendix D: Definitions

Stories
A story is a general term to mean narrative accounts that describe program stakeholder, second-, or third-hand user practice and expose user needs and desires at high levels.

Case Studies
Case studies are explanations or descriptions based on analysis of an entity (or entities) or an event. It is meant to capture an in-depth and potentially long-term view of the case.

Profiles
A profile describes characteristics of an individual member of the defined user community. It summarizes personal work preferences, activities, work habits, tools and artifacts as captured via observation, interview or survey protocols.

Scenarios
Scenarios are brief, narrative **characterizations** that **represent** a merging of direct accounts of activities performed by one or more personas within a user community. While stories articulate a broad vision across a landscape of possibilities and lead to further investigation of specific user needs, scenarios result from those investigations.

Personas

Primary Personas
A primary persona is an archetype of people working toward similar, specific goals within the scope of the product being developed. Personas embody common and critical behavioral characteristics and goals derived from the (often long-term ethnographic) study of people doing real work in actual situations. Each primary persona differs from the others, usually in one critical way. These characteristics form archetypical surrogates, each one representing the mental models, motivations, urgencies, experiences, habits, perceptions, biases, etc. of its set. They are often assigned a personal name to distinguish one from another to reference during communications. Usually there is one primary persona identified in a scenario.

Served Personas (also known as beneficiaries)
Served personas are not actually people who use the system at all, however, they are directly affected by the use of the system. And, notably, served personas are often ultimately the group evaluated to ascertain the success or failure of the product, e.g.; evaluating the level of student learning to see if educational technology available for teachers is effective. These are valuable to identify for summative evaluation activities.

Use Cases

A use case is a methodology used in system analysis to identify, clarify, and organize user and system requirements. The use case contains a set of possible sequences of interactions between a system or set of systems and representative users (known as actors) that take place in a particular context and are related to a particular goal. DC use cases can be system focused or user/science focused and can progress through stages of granularity, refinement and unification. Use cases are an important source of user requirements. Use cases also house metrics or executable tasks by which the use case execution can be measured at iterative phases to evaluate alignment with user needs.

UML notation is to be adopted for the definition of use cases. The template described in Appendix A has been developed and can be used to define all types of use cases developed by both the IS/CS and IRD teams.

Science Use Cases

A Science use case describes the goal of a targeted user, and the requirements of a process that will support scientists to achieve their goals. (E.g., find data on (n) from GenBank and USGS databases.) It is described in technology-free terminology which treats the system as a black box. A science use case will describe a process that provides value to the scientific actor.

In developing science use cases, a process of refinement and unification has been developed and is described in section 5. The different types of science use cases that are generated in this process are defined below.

HIGH LEVEL

A high level use case is one which captures the main goal and includes a basic narrative of the task(s) that need to be performed to achieve the goal.

REFINED

A refined use case is one where the narrative has been decomposed into a detailed set of steps, pre and post conditions have been identified, and the set of actors is clear, i.e. the use case template described in Appendix A has been fully fleshed out. It is anticipated that several iterations and a review process will be required before a refined use case is deemed to be complete.

UNIFIED

A unified use case results from the comparison and potential concatenation of similar use cases across different scientific disciplines. These are mature use cases which are the ultimate basis for requirements and development priorities. There should be a review process in place across IS/CS and IRD to assess readiness of use cases for assignment to this level.

System Use Case

A System use case is a use case that describes an automated function suggested by a science use case. It is normally described at the system functionality level (e.g. "Authenticate User ") and specifies the function or the service that the system provides. The system use case details what the system will do in response to an user's actions. A System Use Case is one that is written from the perspective of system functionality.

Initial Analysis/Formative Evaluation Tasks

Initial analysis and formative evaluation tasks are processes for measuring how successfully a use case has been implemented. While similar, the terms are used in two different camps of system development methodology. Initial Analysis comes from the Unified Modeling Language (UML) world and Formative Evaluation is used in the User Centered Design (UCD) world.

The execution of either may suggest the need for additional investigative research with the user community. There may be multiple tasks for high level and refined use cases, and multiple use cases may be exercised by the same task.

Requirements

USER

A User/Science Requirement is a structured expression of a user need, is expressed from the user's perspective, and meets established criteria. (See Appendix B What Makes a Good Requirement). Multiple user requirements are likely to be identified from a single use case as its sequence of steps is defined. Dependencies of requirements form a hierarchy. Requirements are cross-referenced with their corresponding use case(s).

SYSTEM

A System Requirement is a structured expression of a technical requisite of the system to meet the parameters of a use case. It is expressed from the perspective of the system, can be derived either from a System use case or Science use case and its sequence of steps. Requirements are cross-referenced with their corresponding use case(s).