Anomaly Report Tracking System

User Requirements Document
## Document Status Sheet

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Author</th>
<th>Reason for Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002-06-10</td>
<td>Issue 0.2</td>
<td>Matt Henderson</td>
<td>Internal MakaluMedia Draft</td>
</tr>
<tr>
<td>2002-06-21</td>
<td>Issue 0.3</td>
<td>Matt Henderson</td>
<td>Second Internal MakaluMedia Draft</td>
</tr>
<tr>
<td>2002-06-22</td>
<td>Issue 0.4</td>
<td>Matt Henderson</td>
<td>Third Internal MakaluMedia Draft</td>
</tr>
<tr>
<td>2002-06-28</td>
<td>Issue 1.0</td>
<td>Matt Henderson</td>
<td>Initial Release</td>
</tr>
<tr>
<td>2002-07-25</td>
<td>Issue 1.1</td>
<td>Matt Henderson</td>
<td>ESA comments incorporated</td>
</tr>
<tr>
<td>2002-10-01</td>
<td>Issue 2.0</td>
<td>Matt Henderson</td>
<td>ESA Review and CCN-01</td>
</tr>
<tr>
<td>2003-08-14</td>
<td>Issue 2.1</td>
<td>Phil Sherratt</td>
<td>ESA URD AR's incorporated</td>
</tr>
<tr>
<td>2003-10-13</td>
<td>Issue 2.2</td>
<td>Phil Sherratt</td>
<td>ESA URD AR's incorporated</td>
</tr>
<tr>
<td>2004-09-30</td>
<td>Issue 3.0</td>
<td>Phil Sherratt</td>
<td>ESA CCN4 and CCN5 incorporated</td>
</tr>
</tbody>
</table>

## Document Change Record

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2002-06-10</td>
<td>Issue 0.2</td>
<td>Internal MakaluMedia Draft</td>
</tr>
<tr>
<td>2002-06-21</td>
<td>Issue 0.3</td>
<td>As per internal comments.</td>
</tr>
<tr>
<td>2002-06-22</td>
<td>Issue 0.4</td>
<td>Major revision, based on internal comments and suggestions.</td>
</tr>
<tr>
<td>2002-06-28</td>
<td>Issue 1.0</td>
<td>Initial Release</td>
</tr>
<tr>
<td>2002-07-25</td>
<td>Issue 1.1</td>
<td>Numerous typographical errors corrected</td>
</tr>
<tr>
<td>2002-10-01</td>
<td>Issue 2.0</td>
<td>All changes requested by ESA in their review of Issue 1.1, and all</td>
</tr>
<tr>
<td></td>
<td></td>
<td>changed implied by CCN-01 (Reference [R5])</td>
</tr>
<tr>
<td>2003-08-14</td>
<td>Issue 2.1</td>
<td>The following ESA AR's were incorporated: ESA-12,35,37,69 &amp; 104.</td>
</tr>
<tr>
<td>2003-10-13</td>
<td>Issue 2.2</td>
<td>The following ESA AR's were incorporated: ESA-96,101 &amp; 105</td>
</tr>
<tr>
<td>2004-09-30</td>
<td>Issue 3.0</td>
<td>Incorporated CCN4 and CCN5 as Annexes</td>
</tr>
</tbody>
</table>

Copyright © 2002 by MakaluMedia GmbH
# Table of Contents

**DOCUMENT STATUS SHEET** ........................................................................................................... III

**DOCUMENT CHANGE RECORD** ................................................................................................. III

**TABLE OF CONTENTS** ........................................................................................................ IV

1 **INTRODUCTION** ................................................................................................................... 1
   Purpose of this Document ............................................................................................................ 1
   Acronyms & Abbreviations .......................................................................................................... 1
   Applicable Documents ............................................................................................................... 1
   References ................................................................................................................................. 2

2 **STRUCTURE OF THIS DOCUMENT** .................................................................................. 3

3 **OVERVIEW OF ARTS** ........................................................................................................ 5
   Purpose .................................................................................................................................... 5
   Definitions & Terminology ......................................................................................................... 5

4 **ARTS INSTANCE** ............................................................................................................... 7

5 **SYSTEMS & DELEGATES** ................................................................................................... 8
   Introduction ............................................................................................................................... 8
   Systems Registry ...................................................................................................................... 8
   Default & Foreign Systems ....................................................................................................... 9
   Delegates ............................................................................................................................... 9

6 **USERS** ................................................................................................................................ 11
   Introduction ............................................................................................................................. 11
   User Registry ........................................................................................................................... 11
   Administrator .......................................................................................................................... 11
   Login & Authentication .......................................................................................................... 12
   User Profiles & Communication ............................................................................................ 12

7 **PROJECTS** ....................................................................................................................... 13
   Introduction ............................................................................................................................. 13
   Projects Registry ..................................................................................................................... 13
   Configuration Elements .......................................................................................................... 14

8 **ANOMALY REPORTS** .......................................................................................................... 16
   Introduction ............................................................................................................................. 16
   States of Anomaly Reports ....................................................................................................... 18
   Lifecycle of Anomaly Reports ................................................................................................. 18
   Services .................................................................................................................................. 19
   Classification Hierarchy .......................................................................................................... 19
# ACTIONS

Introduction .................................................................................................................. 21

- 8.1.1 Actions ........................................................................................................... 21
- 8.1.2 Local Actions .................................................................................................. 22
- 8.1.3 Delegate Actions ............................................................................................. 22

States of Actions ........................................................................................................... 23

- 8.1.4 Actions ........................................................................................................... 23
- 8.1.5 Local Actions .................................................................................................. 23
- 8.1.6 Delegate Actions ............................................................................................. 23

Lifecycle of Actions ....................................................................................................... 23

- 8.1.7 Local Actions .................................................................................................. 23
- 8.1.8 Delegate Actions ............................................................................................. 24

# Services

- 25

# ROLES & SERVICES

Overview ....................................................................................................................... 27

- Role Services Hierarchy .......................................................................................... 27
- Supervisors ................................................................................................................ 28
- Deputy Supervisor .................................................................................................... 29
- Originators ................................................................................................................ 29

Actionees ....................................................................................................................... 30

- Guests ....................................................................................................................... 30
- External Actionees (Special Role) ............................................................................. 31

Data Access Services .................................................................................................... 31

- 9.1.1 Pre-Defined Queries ...................................................................................... 31
- 9.1.2 User-Defined Queries .................................................................................... 32

# HISTORIES

- 33

# RELATIONS

- 33

# NOTIFICATIONS

- 34

# ATTACHMENTS

- 36

# COMMUNICATION INTERFACE

Overview ......................................................................................................................... 37

- ACI Messages .......................................................................................................... 37
- 14.1.1 Message Structure ...................................................................................... 37
- 14.1.2 Administrative Header .................................................................................. 38

Services ......................................................................................................................... 39

- Infrastructure .......................................................................................................... 39

# USER INTERFACE

Browser Considerations ................................................................................................. 40

- 15.1.1 Compatibility ................................................................................................. 40
- 15.1.2 Client-Side Technology ................................................................................ 40
- 15.1.3 Layout & Display Techniques ....................................................................... 41
- 15.1.4 Graphics Technologies ................................................................................ 41

Usability Consideration ................................................................................................. 41
16 APPLICATION ENVIRONMENT ................................................................. 44
Platform Infrastructure ............................................................................. 44
Server Infrastructure ................................................................................. 44
Application Infrastructure ......................................................................... 45
Logging 45
Data Backup & Restoration ........................................................................ 46

17 USER REQUIREMENTS ........................................................................ 47

General 47
Systems & Delegates ................................................................................ 47

Users 48
17.1.1 General ............................................................................................ 48
17.1.2 Administrator .................................................................................... 48
17.1.3 Login & Authentication ..................................................................... 48
17.1.4 User Profile & Communication .......................................................... 49

Projects 49
Anomaly Reports ......................................................................................... 49

Actions 50
Roles & Services ........................................................................................ 50
17.1.5 General ............................................................................................ 50
17.1.6 Supervisor ........................................................................................ 51
17.1.7 Deputy Supervisor ............................................................................ 51
17.1.8 Originator .......................................................................................... 51
17.1.9 Actionee ............................................................................................ 51
17.1.10 Guest ................................................................................................. 51
17.1.11 External Actionee ........................................................................... 51
17.1.12 Data Access Services ....................................................................... 51

Histories 52
Notifications ............................................................................................... 52

Attachments ............................................................................................... 53
Communication Interface ............................................................................ 53

User Interface ............................................................................................ 54
17.1.13 Browser Considerations .................................................................. 54
17.1.14 Client-Side Technology ................................................................... 54
17.1.15 Layout & Display Techniques ............................................................. 54
17.1.16 Graphic Technologies ...................................................................... 54
17.1.17 Visual Design .................................................................................. 54
17.1.18 User Interface Considerations ......................................................... 54
17.1.19 Online Help ..................................................................................... 54

Application Environment ........................................................................... 55
17.1.20 Platform Infrastructure .................................................................... 55
17.1.21 Logging ............................................................................................ 55
17.1.22 Data Backup & Restoration .............................................................. 55

18 ANNEX-A (CCN4) .................................................................................. 57
ANNEX-B (CCN5)
1 Introduction

MakaluMedia GmbH has been engaged by the European Space Operations Centre (ESOC) to develop the Anomaly Report Tracking System (ARTS)—a web-based application intended to support the lifecycle of Anomaly Reports (ARs) and their associated Actions.

The original requirements for ARTS were described in three ESA ITT documents, [R1], [R2] and [R3] (see References below.) During the ARTS tender process, the scope of the application to be developed changed, and a number of requirements were modified, as documented in [R4]. Finally, during the project, a CCN was raised to add new requirements and modify existing requirements [R5].

Purpose of this Document

During the first phase of the project, a Requirements Analysis is conducted, with the objective to confirm understanding of the requirements, where applicable or necessary providing clarifications and/or refinements, and harmonising the various original requirements into a single, consolidated, set forming the baseline for the remainder of the project.

There are two outputs to this analysis phase:

- The analysis, clarification and refinement of the requirements captured in [R1] through [R5].
- A new set of harmonised and consolidated User Requirements, found in Section 17 of this document.

The purpose of this document is to capture the outputs of the ARTS Requirement Analysis phase.

Acronyms & Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AR</td>
<td>Anomaly Report</td>
</tr>
<tr>
<td>ARTS</td>
<td>Anomaly Report Tracking System</td>
</tr>
<tr>
<td>CI</td>
<td>ARTS Communication Interface</td>
</tr>
<tr>
<td>UR</td>
<td>User Requirement</td>
</tr>
</tbody>
</table>

Applicable Documents

None.
References


[R6] ARTS Layout_2.pdf (Email from Stefano Scaglioni on 2003-11-21.)
Structure of this Document

The structure of this document is as follows:

- **Overview of ARTS.** The Overview of ARTS section introduces the context and intended functionality of the system to be developed, and defines relevant terminology.

- **ARTS Instance.** The ARTS Instance section defines the logical components of an ARTS installation.

- **Systems & Delegates.** The Systems & Delegates section describes how external systems and their Delegates (i.e. Projects to which Actions may be delegated) are managed.

- **Users.** The Users section describes the management of ARTS Users, and discusses system access (login) requirements.

- **Projects.** The Projects section details the management and configuration of ARTS Projects.

- **Anomaly Reports.** The Anomaly Reports section details the structure, states, lifecycle and services associated with ARs.

- **Actions.** The Actions section details the structure, states, lifecycle and services associated with Actions.

- **Roles & Services.** The Roles & Services section defines the various user roles supported by ARTS, and the services that ARTS provides these users.

- **Histories.** The Histories section introduces the History logical element, and the role it plays in capturing information related to Anomaly Reports and Action.

- **Relations.** The Relations section introduces the Relation logical element, and the role it plays in establishing relationships between two Anomaly Reports.

- **Notifications.** The Notifications section explains the ARTS concept of email notifications, both to users involved in the lifecycle of ARs and Actions, and to external users interested in the projects supported by ARTS.

- **Attachments.** The Attachments section explains how ARTS supports the association of files to ARs and Actions.

- **Communication Interface.** The Communication Interface section describes how ARTS can involve Delegates in the lifecycle support of Actions, through email-based messages passed on its communication interface.

- **User Interface.** The User Interface section discusses usability and visual design issues, and a number of considerations and contraints related to the fact that ARTS is accessed through a common web browser.
• **Application Environment.** The Application Environment section discusses considerations and requirements related to the environment in which ARTS is operated.

• **User Requirements.** The User Requirements section of the document contains the specific, numbered User Requirements applicable to the remainder of the ARTS project.
2 Overview of ARTS

Purpose

The purpose of ARTS is to support the lifecycle of Anomaly Reports (ARs) and their associated Actions, as outlined in [R2], providing the following services:

1. Generation and storage of Anomaly Reports associated to Projects by Originators
2. Acceptance or rejection of newly submitted Anomaly Reports by a Supervisor
3. Generation of Actions associated to Anomaly Reports.
4. Assignment (or delegation) of Actions by a Supervisor to Actionees (or Delegates).
5. Responding to Actions by Actionees or Delegates.
6. Processing of Action responses by a Supervisor.
7. Closure of Anomaly Reports by a Supervisor.

These basic services are supplemented by system services supporting the process of managing Anomaly Reports:

8. Listing of Anomaly Reports. Anomaly reports may be listed according to queries depending on the role of the user of the system.
9. Browsing of Anomaly Reports. Anomaly Reports may be browsed from a list, which has either been generated by a pre-defined query or a user-defined query.

In order to support the delegation of Actions to other Projects within or outside an ARTS installation, ARTS provides an external Communication Interface, based on the exchange of email messages.

Definitions & Terminology

Instance

An ARTS Instance refers to an installation of the ARTS application on a single host machine. One ARTS Instance can support any number of ARTS Projects, limited only by the physical resources (memory, disk space, etc.) of the host platform.

Database

An ARTS Database refers to the data store supporting an ARTS installation. This will typically correspond to the MySQL database defined for the ARTS installation.
Delegate
An entity, other than a local User, to which a Project Action may be assigned (delegated). This typically would include another Project within the Instance, or a Project within a Foreign System.

Project
An ARTS Project refers to the data and services associated to a single project defined within an ARTS Instance. This is often referred to as a database instance in the ESA project documentation.

Owner
The ARTS Owner is the person or persons with administrative access to the ARTS-related files, data and scripts on host platform of the ARTS Instance. Such a person would typically be a Unix System Administrator. The Owner does not correspond to a Role within ARTS, and indeed ARTS provides no services to the Owner.

Role
An ARTS User may be assigned to one ARTS Role per Project — Supervisor, Deputy Supervisor, Originator, Actionee, External Actionee or Guest.

User
An ARTS User is defined in association with an ARTS Instance, and may be assigned to Roles within one or more ARTS Projects within that Instance.

Foreign System
Any external system that supports message exchange compatible with the ARTS Communication Interface.

Local Action
An Action that was created and assigned within the same ARTS Project. Local Actions are also simply referred to as Actions.

Delegate Action
An Action that was delegated to an ARTS Project, but was created outside that project.

Foreign AR
An Anomaly Report that was created from the information delivered with a Delegate Action.
3 ARTS Instance

An ARTS Instance represents a unique installation of the ARTS application on a host server machine that contains an operational environment (described in detail in Section 16) suitable for the running of the application. Users access the application with a web-browser.

The installation of an ARTS Instance is supported by installation scripts that, when run by the Owner, create needed resources and set the initial configuration of certain elements.

Configuration of an ARTS Instance is also performed by the Owner (or anyone else with sufficient access) through the setting of parameters in text files located in the host server’s file system.

ARTS contains a number of application components that are logically related to the Instance, and which are configured and managed by the ARTS Administrator (who, in most cases, would likely also be the Owner):

- **Projects Registry.** The Projects Registry (described in Section 0) contains the Projects that are supported by the particular ARTS Instance. It also contains the Role association of Users to Projects, and the association of Delegates to the Projects.

- **User Registry.** The User Registry (described in Section 0) contains the Users that are allowed to access the particular ARTS Instances.

- **Systems Registry.** The Systems Registry contains the registration of Systems that have announced the availability of Delegates to which actions may be delegated.

- **Data Backup & Maintenance.** The Data Backup & Maintenance component contains backups of the ARTS Instance’s database contents stored in ASCII text files on the host servers’s file system.

These components, and the related management services provided to the Administrator, are described in subsequent sections of this document.

The creation of the Administration User is discussed in Section 0.
4 Systems & Delegates

Introduction

A key aspect of ARTS is the ability to delegate Actions. As opposed to assignment to a User, Actions associated to a particular Project within an Instance may be delegated to other Projects within the same Instance, or to Projects located within known Foreign Systems. In either case, the Project to which an Action has been delegated is known as a Delegate.

Systems Registry

Actions within an ARTS Instance may only be delegated to Delegates associated with Systems existing within the Instance’s Systems Registry. There are two types of Systems defined in the Systems Registry:

- **Default System.** The Default System is the ARTS Instance itself. As an entry in the Systems Registry, Projects within an Instance can delegate Actions to other Projects within the same Instance.

- **Foreign System.** Any external system that is compatible with the ARTS Communication Interface.

Each entry in the Systems Registry contains administrative information, and information about the available Delegates (defined in Sections 0 and 0, respectively). The installation scripts initialise the Default System entry when the Instance is first installed.

The Systems Registry provides services to the Administrator to:

- manage the Default System information (see following section)

- announce or update the availability of the Instance (Default System) to other systems, so that its Projects can act as Delegates for those systems.

- manage the automated creation, automated update and manual deletion of Foreign System entries in the registry. A Foreign System may not be deleted if there are associated delegated actions in the Remote In-Progress or the Remote Responded state.

Announcement or update of the Instance’s availability to other external systems occurs through the transmission of an ACI Message passed on the ARTS Communication Interface (described in Section 14). Such an announcement can be sent to a Foreign System registered in the Systems Registry, or to an email address manually entered by the Administrator.

Requests to create or update a Foreign System entry in the Registry occurs through the reception of an ACI Message passed on the ARTS Communication Interface (described in Section 14). It is not
possible for the Administrator to manually create or update a Foreign System entry, although it is possible for him to manually delete an entry.

ARTS does not provide services to ensure the consistency at any given time between information about an entry in the Systems Registry and the actual current state of the Foreign System.

**Default & Foreign Systems**

Each System (Default & Foreign) entry in the Systems Registry contains the following information:

- **System ID.** In order to help ensure the unique identification of an ARTS Instance, the scripts that support the ARTS Installation creates a numeric identifier based on a timestamp and a random number. This identifier becomes the **System ID.**

- **Short Name.** A maximum eight-character name identifier for the ARTS Instance (e.g. ESOC2, ESTEC3, etc.)

- **Long Name.** A maximum 255-character name identifier for the ARTS Instance (e.g. ESOC Flight Dynamics)

- **Description.** A maximum 64 kB description of the ARTS Instance.

- **URL.** The Internet address at which the ARTS is accessible (e.g. http://arts.esoc.esa.int/)

- **CI Address.** The email address to which Communication Interface (described in Section 14) are sent.

- **Administrator Name.** The name of the Instance Administrator.

- **Administrator Address.** The email address of the Instance Administrator.

- **Delegates.** A list of delegates that may be assigned with Actions.

**Delegates**

A Delegate refers to Project within a Foreign System to which Actions in the Local System can be delegated. All Local Projects are available for Action delegation locally, and are therefore not registered in the Delegate Registry.

Each Delegate entry in the Systems Registry contains the following information:

- **ID.** This corresponds to the Project ID of the delegate Project on the Foreign System.

- **System ID.** A maximum eight-character name identifier for the Delegate (e.g. IFMS, NCTRS, etc.)

- **Delegate Name.** A maximum eight-character name identifier for the Delegate (e.g. IFMS, NCTRS, etc.)
• *Long Name.* A maximum 255-character name identifier for the Delegate (e.g. Intermediate Frequency & Modem System)

• *Description.* A maximum 64 kB description of the Delegate.

• *Supervisor Name.* The name of the Delegate’s Supervisor.

• *Supervisor Address.* The email address of the Delegate’s Supervisor.

• *Project Associations.* A list of associations between the Delegate and the local Projects for the purpose of Action delegation.
5 Users

Introduction

An ARTS User is a person that is allowed to access an ARTS Instance, for the purpose of participating in a particular Role within one or more Projects (See Section 6 for more information on Projects, and Section 9 for more information on Roles, or permission levels, within Projects.)

Each User is defined by the following information:

- **User ID.** A non-modifiable, automatically created numeric attribute that uniquely identifies the User within the Instance.

- **Username.** An identifier, unique to the ARTS Instance, provided for the purpose of login. By convention, the system shall equate the username to the User’s email address.

- **Password.** A maximum eight-character password required for login to the Instance.

- **Last Name.** The User’s family name.

- **First Name.** The User’s surname.

- **Email Address.** The User’s email address.

There is one special User associated to each ARTS Instance, the **Administrator** (described in Section 0).

User Registry

The ARTS Instance’s central User Registry contains one entry for each defined User. ARTS provides the Administrator with all services to manage the User Registry:

- management (creation, update and deletion) of Users

Administrator

As discussed throughout this document, one special User exists, known as the **Administrator**. The Administrator is a regular ARTS User who is identified in the ARTS configuration (email address and password) as having administration privileges. The Administrator User is initialised automatically during installation.

In addition to the services necessary to administer and manage the ARTS Instance (described in various places throughout this document), the Administrator is considered to be a **Superuser** in
terms of Roles (see Section 9) within Projects—meaning that within the context of a Project, he can at any time perform any action available to any user within that Project.

The User identified in the Instance configuration to be Administrator may not be deleted from the User Registry.

**Login & Authentication**

Since ARTS may provide access to sensitive information, all users must be registered and log into the system before they can use ARTS services. Login requires a valid username and password.

Upon successful user authentication (login) ARTS starts a session, which will be maintained until the user logs out or is inactive for a configurable period of time. ARTS will allow concurrent sessions of the same user. ARTS will allow the user to logout at any time.

Logout Services will not be displayed on pop up windows created by ARTS.

Upon successful user authentication, ARTS presents the User with a Welcome Page listing links to all Projects (if any) to which he is associated. See Section 6 for more information about Projects.

At any time the user may return to the Welcome Page (without logging out), thereby allowing within the same session access to multiple projects (and therefore possibly multiple Roles, as described in Section 9).

The ARTS login screen provides the service to all users to have their password sent to their registered email address. This service allows retrieval in the case of lost or forgotten passwords.

ARTS shall allow unregistered users to communicate with the Administrator via a form-based email.

**User Profiles & Communication**

Within the ARTS User Interface, Users are presented as links to a User Profile page that:

- Displays administrative information about the User (name, email address, etc.)
- Lists the Projects to which they are associated, and the Roles assigned within those Projects.
- Provides a form for the purpose of writing and transmitting an email directly to the User from within ARTS.

ARTS shall provide the service to all Users to view their own profile. When a User views his own profile, the administrative data is presented in a form, which allows the User to modify his profile.
6 Projects

Introduction

A Project is a logical component within an ARTS Instance for the purpose of grouping related Anomaly Reports and Actions. In general, an ARTS Project would typically correspond to an existing ESOC project in which ARs and actions are raised, such as the IFMS. An ARTS Project may additionally act as a Delegate for the purpose of action delegation by other ARTS Projects or Foreign Systems (see Section 4 for more information on Delegates).

All Projects supported by an ARTS Instance are defined by the Administrator in the Instance’s Projects Registry. Each Project entry in the Projects Registry contains the following information:

- **Project ID.** A non-modifiable, automatically created numeric attribute that uniquely identifies the Project within the Instance.
- **Short Name.** A maximum eight-character name identifier for the Project (e.g. IFMS, NCTRS, etc.)
- **Long Name.** A maximum 255-character name identifier for the Project (e.g. Intermediate Frequency & Modem System)
- **Description.** A maximum 64 kB description of the Project.

Projects are accessed by ARTS Users that have been granted particular permission levels (Roles), such as Supervisor, Deputy Supervisor, Originator, Actionee, External Actionee, or Guest. The assigned permission level, or Role, determines the services that are available to the user within the Project. A Project must be assigned one (and only one) Supervisor. It may be assigned one or more Deputies, Originators, Actionees, External Actionees and Guests. See Section 9 for further description of Roles.

Projects Registry

An ARTS Instance’s Project Registry contains one entry for each Project supported within the Instance and provides services to the Administrator to:

- create new projects, and manage (modify) existing Project’s administrative information (Short Name, Long Name & Description)
- manage the association (assignment, removal) of Project Roles to Instance Users. A Role may not be removed from a User if there are active Anomaly Reports or Actions associated to the User within the Project. (In such a case, the Administrator would be required to access the
Project in his capacity as a superuser, and manually bring to completion any Anomaly Report or Action.)

- delete a Project from the Instance. The deletion of a Project removes from the Instance all data and files associated with the Project, including Anomaly Reports, Actions and User Role associations. A Project may not be deleted that contains active Anomaly Reports and Actions. (In such a case, the Administrator would be required to access the Project in his capacity as a superuser, and manually bring to completion all Anomaly Reports and Actions.)

- manage the association of Delegates to local Projects for the purpose of Action delegation

Configuration Elements

Several logical elements associated to a Project must be manually defined/configured by the Instance Administrator.

The following configuration element must be manually defined through the modification of XML-like text files on the Instance’s host platform:

- Classification Hierarchy. The classification structure into which ARs are associated. (See Section 0)

While ARTS provides no services for the management of Classification Hierarchy, it does validate the external configuration file’s syntax. Errors detected are reported to the Administrator in the Projects Registry.

ARTS provides services within the Projects Registry to defined and modify the following configuration elements:

- AR Criticality. The list of values for which the AR Criticality attribute may assume. (See Section 7)

- AR Urgency. The list of values for which the AR Urgency attribute may assume. (See Section 7)

- AR Report Type. The list of values for which the AR Report Type attribute may assume. (See Section 7)

- AR Event Type. The list of values for which the AR Event Type attribute may assume. (See Section 7)

- AR Reproducibility. The list of values for which the AR Reproducibility attribute may assume. (See Section 7)

- AR Time Reference. The time reference for the creation time of new Anomaly Reports, either System or UTC. (See Section 7)
In the case of simple listings, e.g. AR Criticality, AR Urgency, AR Report Type, AR Event Type and AR Reproducibility, list elements are defined in the Project Registry and separated by the comma character, and the first list element is entered as the default unless the User chooses otherwise.
7 Anomaly Reports

Introduction

In ARTS, an Anomaly Report (AR) is an entity containing a report by a User, with Originator privileges or higher (see Section 9), of an anomalous event related to a Project. The structure of an AR is as follows:

- **AR Instance ID.** A non-modifiable, automatically created numeric attribute that uniquely identifies the AR within the Instance.

- **AR ID.** A non-modifiable, automatically created numeric attribute that uniquely identifies the AR within the Project. The AR ID, as opposed to the AR Instance ID, allows all ARs within a project to have a contiguous sequential numbering.

- **Project ID.** The identifier of the Project to which the AR is associated.

- **History ID.** The identifier of the History entity associated to the AR. (See Section 10 for more information about Histories.)

- **State.** Represents the logical state of an AR in its lifecycle. (The definition and meaning of AR states are defined in the following section.)

- **Date.** The creation date of the AR, in the format YYYY-MM-DD.

- **Time.** The creation time of the AR, in the format HH-MM-SS corresponding to the host server time settings and zone.

- **Day-of-Year.** The day of the year (1-365) derived from the creation date of the AR.

- **Originator ID.** The identification of the User creating the AR.

- **Originator Initials.** The initials of the Originator, derived from his Surname and Family Name.

- **Classification Level 1.** The first level of AR hierarchial classification. In ESOC this would typically correspond to Segment. (See Section 0 for more on Classification Hierarchy.)

- **Classification Level 2.** The second level of AR hierarchial classification. In ESOC this would typically correspond to Element. (See Section 0 for more on Classification Hierarchy.)

- **Classification Level 3.** The third level of AR hierarchial classification. In ESOC this would typically correspond to Subsystem. (See Section 0 for more on Classification Hierarchy.)

- **Related ARs.** A listing of the AR Instance IDs of ARs related to this AR.
• **Attachments.** A listing of files that have been uploaded and associated to this AR.

• **Observation.** A maximum 255 character title of the AR.

• **Description.** A maximum 64 kB description of the AR.

• **Criticality.** An indication of the severity of the anomaly. The possible values of the Criticality are defined through configuration on a Project basis (see Section 0).

• **Urgency.** An indication of the urgency in which the anomaly needs to be addressed. The possible values of the Urgency are defined through configuration on a Project basis (see Section 0).

• **Environment.** An optional, maximum 64 kB, text describing the conditions under which the anomaly has been detected.

• **Report Type.** An optional indication of the type of anomaly report. The possible values of the Report Type are defined through configuration on a Project basis (see Section 0).

• **Event Type.** An optional indication of the type of anomalous event. The possible values of the Event Type are defined through configuration on a Project basis (see Section 0).

• **Recommendation.** The optional, maximum 64 kB, text describing the originator’s recommended course of action.

• **Item Configuration.** The optional, maximum 64 kB, description of the version of the item affected by the anomaly.

• **Affected Requirement.** The optional, maximum 64 kB, description of the requirement identifier and relevant document associated to the anomaly.

• **Reproducibility.** An optional indication of the extent to which the anomalous event is reproducible. The possible values of the Reproducibility are defined through configuration on a Project basis (see Section 0).

• **Resolution Due Date.** The optional due date, in YYYY-MM-DD format, as recommended by the ARs originator.

• **Project Authority.** The User ID of the Supervisor of the AR’s associated Project.

• **Root Cause Analysis.** An optional, maximum 64 kB, identification of the cause(s) leading to the reported anomaly.

• **Preventive Actions.** An optional, maximum 64 kB, indication of the actions taken after the resolution of the anomaly to prevent similar occurrences in the future.

• **Resolution.** The optional, maximum 64 kB, Project Authority’s final comments on the AR lifecycle.
The Notification Date of an AR (as described in [R2]) is captured in the ARs associated History entity. (See Section 10 for more information about Histories, and Section 12 for more information about Notifications.)

**States of Anomaly Reports**

The following are the defined states of an Anomaly Report:

- **Open.** An AR that has been created, but has not been assessed.
- **Pending.** An AR that has been assessed, and is in the process of resolution. (This would typically imply, but would not require, that one or more associated Actions have been created.)
- **Testing.** An AR whose resolution awaits assessment.
- **Resolved.** An AR whose resolution has been accepted and awaits closure.
- **Closed.** An AR that has been closed with a satisfactory resolution.
- **Rejected.** An AR that has been otherwise closed.

**Lifecycle of Anomaly Reports**

Each Anomaly Report undergoes a typical management process from its origin to its end, as described below. The description refers to User permission levels (Roles), which are defined and further described in Section 9.

1. The lifecycle begins with the detection of a problem and the generation of an Anomaly Report, by a User with at least Originator level of permission in the Project. The AR state is set to Open as soon as it is created.

2. Newly generated ARs are assessed by the Supervisor and the state is then either changed to Pending or to Rejected depending on the result of the assessment (rejection implies a closing of the AR).

3. If the state was set to Pending, the Supervisor may assign an Action to the AR and processing is passed on to an Actionee (or any other user in the Project with at least Actionee level of permission) or to a Delegate. (More than one Action may be created, based on the same parent AR.) The Actionee or Delegate investigates or solves the problem associated with the AR and reports the results to the Supervisor. Depending on the results this cycle may be repeated several times (possibly with different assignments) until the Action is completed.

4. When all Actions have been completed, the state of the AR is set by the Supervisor to Testing, and the AR is passed to the Originator with a request to test the solution.
5. If the Originator reports successful testing by setting the ARs state to Resolved, the Supervisor then closes the AR by setting its state to Closed. Otherwise, if the test failed, the AR state is set by the Originator to Open and processing goes back step 1. (An AR may not be closed while there remain outstanding Actions.)

In contrast to Anomaly Reports, it should be noted that a Foreign Anomaly Report is simply a collection of static information associated to a Delegate Action, and as such differ from Anomaly Reports in that they do not have states or a lifecycle.

**Services**

Depending on both the state of the AR, and the permission level (Role) of the accessing User, ARTS provides the following services related to ARs:

- creation of new ARs
- modification of existing ARs
- deletion of existing ARs

The particular fields that are modifiable in a given state depend on the Role of the user accessing the AR, and shall be agreed between ESA and MakaluMedia during the detailed design phase.

The deletion of an AR initiates the following automatic actions:

- deletion of all data and files associated with the AR
- deletion of any Actions associated with the AR
- deletion of the associated History entity (see Section 10)
- deletion of any associated Relation entities (see Section 11)

Only ARs in the Closed or Rejected state may be deleted.

**Classification Hierarchy**

Each Anomaly Report may be classified within a three-level Classification Hierarchy. In the case of an ARTS Instance at ESOC, these three levels would typically correspond to Segment, Element and Subsystem.

The contents of these levels are defined on a per-Project basis, in ASCII text files on the host server. The Classification Hierarchy data shall be encoded in XML data structures and shall defined:

- The Level 1 title (e.g. Segment)
- All Level 1 entries
• The Level 2 title (e.g. Element)

• All Level 2 entries for each Level 1 entry

• At most one Notification List identifier per Level 2 entry (See Section 12 for more information about Notifications.)

• The Level 3 title (e.g. Subsystem)

• All Level 3 entries for each Level 2 entry

ARTS shall not provide services for the management of Classification Hierarchy definitions, but shall provide validation of the definition’s syntax and contents. Errors shall be reported to the Administrator in the Projects Registry.
8 Actions

Introduction

In ARTS, an Action is an entity representing an AR-related task that has been assigned to a User who has been granted at minimum Actionee permission level (Role) in the Project (see Section 9 for more on Project Roles, or permission levels), or that has been delegated to a Delegate Project (see Section 4 for more on Delegates).

Within any Project, there are two types of Actions that may exist: Local Actions and Delegate Actions. An action that has been created locally, and is therefore associated to a local Anomaly Report, is known as a Local Action. An action that has been created due to a delegation from another Project or Foreign System, and is associated to a Foreign Anomaly Report, is known as a Delegate Action. The commonalities between a Local and Delegate Action are captured in a transparent entity known simply as an Action.

Important: It should be noted that outside of this and the following section of the document, Local Actions are simply referred to as Actions.

8.1.1 Actions

The structure of an Action is as follows:

- **Action Instance ID.** A non-modifiable, automatically created numeric attribute that uniquely identifies the Action within the Instance.

- **Action ID.** A non-modifiable, automatically created numeric attribute that uniquely identifies the Action within the Anomaly Report. The Action ID, as opposed to the Action Instance ID, allows all Actions within a Anomaly Report to have a contiguous sequential numbering.

- **AR Instance ID.** The identifier of the AR to which the Action is associated.

- **Project ID.** The identifier of the Project to which the associated AR is associated.

- **History ID.** The identifier of the History entity associated to the Action. See Section 10 for more information about Histories.

- **Assignment.** This indicates whether an Action is Local, Delegate or None. (Delegate is only for Local Actions).

- **Assignment ID.** This refers to a local actionee if the Assignment is set to Local, Delegate if Assignment is set to Delegate and None if the Action is Unassigned. (Delegate is only for Local Actions)
- **State.** Represents the logical state of an Action in its lifecycle. The definition and meaning of Action states are defined in the following section.

- **Date.** The creation date of the Action, in the format YYYY-MM-DD.

- **Start Date.** The specified start date of the Action, in the format YYYY-MM-DD.

- **Due Date.** The specified due date of the Action, in the format YYYY-MM-DD.

- **Title.** A maximum 255 character title of the Action.

- **Description.** A maximum 64 kB description of the Action.

- **Attachments.** A listing of files that have been uploaded and associated to this Action.

- **Solution Configuration.** An optional, maximum 255 character, text containing the version of the item affected by the anomaly after the action solution has been implemented.

- **Solution Integrator.** An optional, maximum 255 character, text containing the name(s) of the person(s) implementing the action solution.

- **Solution Test.** An optional, maximum 64 kB, text containing the test identifier and documents used for testing of the action solution.

- **Supplier Reference.** An optional, maximum 255 character, text containing a Supplier reference to the action. In the case of an Local Action that was delegated, this field would automatically contain the identifier of the Delegate Action on the Delegate system.

- **Customer Reference.** An optional, maximum 255 character, text containing a Customer reference to the action. In the case of an Delegate Action, this field would automatically contain the identifier of the Local Action on the originating system.

The Response to an Action (as described in [R2]) is captured in the Action’s associated History entity. (See Section 10 for more information about Histories.)

### 8.1.2 Local Actions

The structure of a Local Action is equivalent to an Action, although it assumes additional possible states, as described in Section 8.1.5.

### 8.1.3 Delegate Actions

The structure of a Delegate Action is derived from an Action, and extended with the following attributes:

- **Foreign System ID.** The System ID of the originating system.

- **Foreign Project ID.** The Project ID of the associated Local Action on the originating system.
- **Delegated Action ID.** The Action Instance ID of the associated Local Action on the originating system.
- **Delegated Action History.** The contents of the History of the associated Local Action on the originating system.
- **Delegated Action Date.** The Date of the associated Local Action on the originating system.

**States of Actions**

**8.1.4 Actions**
The following are the defined states of an Action:
- **Unassigned.** An action that is presently not assigned or delegated.
- **In-Progress.** An action that has been assigned to a local User within the Instance.
- **Responded.** An action that has been responded to by a local User and awaits assessment.
- **Completed.** An action that has been closed with a satisfactory response.
- **Rejected.** An action that has been otherwise closed.

**8.1.5 Local Actions**
The states of a Local Action include the states of an Action, extended to include:
- **Remote In-Progress.** An action that has been delegated to a Delegate.
- **Remote Responded.** An action that has been responded to by a Delegate and awaits assessment.

**8.1.6 Delegate Actions**
The states of a Delegate Action include the states of an Action, extended to include:
- **Remote Validation.** A Delegate Action whose response, considered locally acceptable, has been returned to the originating system for assessment.

**Lifecycle of Actions**

**8.1.7 Local Actions**
Each Local Action undergoes a typical management process from its origin to its end, as described below. The description refers to User permission levels (Roles), which are defined and further described in Section 9.
1. The lifecycle begins with the creation of an Action, associated to a parent Anomaly Report, by a Supervisor. The initial state of an Action is Unassigned.

2. The Supervisor may assign an Unassigned Action to a local Project User with at least Actionee permission level, in which case the state of the Action is set to In-Progress, or he may delegate the action to a Delegate that has been associated to the Project, in which case the state of the Action is set to Remote In-Progress.

3. When a Delegate Action is in the Remote In-Progress state it can still be updated by the Supervisor. The updates will be sent to the Foreign System on selecting and only on selecting the Update Action button. Any other changes made to the action such as deletion/creation of attachments or changes in the associated Anomaly Report will not be sent to the Foreign System (i.e. will only be stored locally) until the Update Action button has been selected.

4. An Actionee, while processing an assigned Action, may add information to the Action prior to a final response, and without affecting the state of the Action. Once an Actionee responds to an Action, its state is set to Responded, at which point it returns to the Supervisor for assessment. Once a Delegate responds to an Action, its state is set to Remote Responded, at which point it returns to the Supervisor for assessment.

5. Upon assessment of the Responded or Remote Responded Action, the Supervisor may return the Action to the (respective) Actionee or Delegate for further processing if he is unsatisfied with the response. If he is satisfied with the response, he can close the Action by setting its state to Completed. He additionally has the authority to close the Action by setting its state to Rejected, in the case that he is unsatisfied with the response to the Action but does not wish further processing. (A state of Rejected implies a closure of the Action.)

8.1.8 Delegate Actions

Each Delegate Action undergoes a typical management process from its origin to its end, as described below. The description refers to User permission levels (Roles), which are defined and further described in Section 9.

1. A Delegate Action Message is received from a Project in the Project ACI area. The Delegate Action Message can be accepted by the Supervisor of the Project that received the Action Message or the Action can be Rejected before it enters the local project as a Delegate Action by selecting the Action Reject button and indicating why the Action Delegation has been Rejected in a comment field. Rejection of an incoming Action Delegate Message is done by sending an Action Update Reject ACI Message to the originating system. A rejected Delegated Action Message does not enter the local project and therefore does not enter the Delegate Action lifecycle.

2. The lifecycle begins with the acceptance and creation by a Supervisor of a Delegate Action, based on a request (ACI Message, see Section 0) received from another Project or Foreign System containing a delegated Action. The initial state of the Delegate Action is set to
3. The Supervisor assigns an Unassigned Delegate Action to a local Actionee (or any other user with at least Actionee permission level in the Project), in which case the state of the Action is set to In-Progress. (The state of the corresponding Local Action on the originating Project or Foreign System remains Remote In-Progress.)

4. The Actionee, while processing the assigned Action, may add information to the Action prior to a final response, and without affecting the state of the Action. Once the Actionee responds to the Action, its state is set to Responded, and which point it returns to the Supervisor for assessment. (The state of the corresponding Local Action on the originating Project or Foreign System remains Remote In-Progress.)

5. Upon assessment of the Responded Action, the Supervisor may return the Action to the Actionee for further processing, if he is unsatisfied with the response. If he is satisfied with the response, he can transmit the response to the originating Project or Foreign System through an ACI Message, at which point the state of the action is set to Remote Validation. (The state of the corresponding Local Action on the originating Project or Foreign System is set to Remote Responded.)

6. If the originating Project or Foreign System considers the response satisfactory, an ACI Message is received with a request to the Supervisor to allow the state of the Delegate Action to be set automatically set to Completed. If the originating Project or Foreign System considers the response unsatisfactory, an ACI Message is received with a request to the Supervisor to allow the state of the Delegate Action to be set to Unassigned. (It is not desired that the originating Project or Foreign System is allowed to directly return the Delegate Action to the local Actionee.) Finally, the originating Project or Foreign System also has the authority to request to the Supervisor that the Delegate Action state is automatically set to Rejected, implying a closure.

Services

Depending on both the state of the Action, and the permission level (Role) of the accessing User, ARTS provides the following services related to Actions:

- creation of new Actions
- modification of existing Actions
- deletion of existing Actions

The particular fields that are modifiable in a given state depend on the Role of the user accessing the Action, and shall be agreed between ESA and MakaluMedia during the detailed design phase.

The deletion of an Action initiates the following automatic actions:
- deletion of all data and files associated with the Action
- deletion of the associated History entity (see Section 10)

Only Actions in the Completed or Rejected state may be deleted.
9 Roles & Services

Overview

The process of managing Project-related ARs and Actions throughout their lifetime involves certain Roles of the people participating in this process.

- Supervisor
- Deputy Supervisor
- Originator
- Actionee
- Guest
- External Actionee (a special role)

The following sections provide an overview of the services provided by ARTS to each Project-related Role in the support of AR and Action nominal lifecycles. (Exceptional services provided on a role-by-role basis will be later specified in the detailed design of the system.)

The assignment of Users to Roles within Projects is a service provided to the Administrator within the Instance’s Project Registry (see Section 0).

Role Services Hierarchy

With exception of External Actionee, project-related roles—Supervisor, Deputy Supervisor, Originator, Actionees and Guests—are hierarchical, in that all services propagate upwards. For example, a Supervisor can act as an Originator and an Actionee. An Originator can act as an Actionee. An Actionee, however, cannot act as an Originator.

This implies that each Project-related Role assigned to a User represents a particular level of permission to access services provided by ARTS related to ARs and Actions.

In this way, it is only necessary to assign one Role per User per Project, and will allow users to access all desired and permitted services without having to logout of a Project. For example, if a Supervisor wishes to create an Anomaly Report and assign himself with an Action, he can do so without having to logout.
Supervisors

The Supervisor of an ARTS Project is the ARTS User responsible for lifecycle decisions on Anomaly Reports and Actions within the Project. There is one (an only one) Supervisor defined per Project.

The Supervisor defines the team involved in the anomaly management and identifies the Roles of those people (Originators, Actionees & Guests) in the context of his Project, along with other project-relevant information (e.g. the structuring into segments, elements, and subsystems). He provides this information to the Administrator for configuration of the Project in ARTS.

To facilitate communication with the Administrator, ARTS provides the Supervisor with a form to generate an email to the Administrator, via the Administrator’s User Profile Page (see Section 0).

Specifically, ARTS provides the Supervisor with the following services within his Project:

- All services available to an Originator on ARs created by himself
- All services available to an Actionee on Actions assigned to himself
- Ability to set Open ARs to Pending or Rejected
- Ability to set Pending ARs to Testing or Rejected
- Ability to set Testing ARs to Pending or Rejected
- Ability to set Resolved ARs to Pending, Closed or Rejected
- Ability to modify AR information in all states. (The particular information—i.e. fields—that may be modified depend on the state and will be agreed between MakaluMedia and ESA during the detailed design phase.)
- Ability to delete Closed and Rejected ARs
- Ability to create new Unassigned Actions
- Ability to set Unassigned Actions to In-Progress (by assigning them to a local Users), or to Remote In-Progress (by delegating them to Delegates), or to Rejected.
- Ability to set In-Progress Actions to Unassigned.
- Ability to set Remote In-Progress Actions to Unassigned or (on request of a Delegate system) to Remote Responded.
- Ability to set Respondered Actions to Unassigned, In-Progress, Completed or Rejected.
- Ability to set Remote Responded Actions to Unassigned, Remote In-Progress, Completed or Rejected.
• Ability to modify Action information in all states. (The particular information—i.e. fields—that may be modified depend on the state and will be agreed between MakaluMedia and ESA during the detailed design phase.)

• Ability to create new Unassigned Delegate Actions (on request of an originating system).

• Ability to set Unassigned Delegate Actions to In-Progress (by assigning them to an ARTS User) or to Rejected (on request of an originating system, or on self-initiative).

• Ability to set In-Progress Delegate Actions to Unassigned, or to Rejected (on request of an originating system, or on self-initiative).

• Ability to set Responded Delegate Actions to Remote Validation or to Rejected (on request of an originating system, or on self-initiative).

• Ability to set Remote Validation Delegate Actions to Completed (on request of an originating system) or to Rejected (on request of an originating system, or on self-initiative).

• Ability to delete Completed and Rejected Actions.

• Received email-based notifications of relevant ARTS events. (The particular events for which notifications will be sent will be defined and agreed between MakaluMedia and ESA during the detailed design phase.)

• All AR and Action browsing and viewing services, as specified in Section 0.

Deputy Supervisor

The Deputy Supervisors within an ARTS Project has all privileges of a Supervisor in the lifecycle management of ARs and Actions. There may be multiple Deputy Supervisors defined per Project. Given the similarity between the services available to the Deputy Supervisor and the Supervisor, the following list defines the differences between the roles:

• The Supervisor receives all communications related to the lifecycle of ARs and Actions within his Project. By default, the Deputy Supervisor only receives communications related to interactions he specifically initiates in the lifecycle of an AR or Action.

• It shall be possible to enable for any particular Deputy Supervisor, to receive all notifications related to the lifecycle of ARs and Actions within a Project (but excluding Communication Interface related notifications.) In this way, aside from Communication Interface notifications, the Deputy Supervisor is identical to a Supervisor.

Originators

The Originators within an ARTS Project are the ARTS Users responsible for the detection and reporting of problems. There may be multiple Originators defined per Project.
Specifically, ARTS provides the Originator with the following services:

- All services available to an Actionee on Actions assigned to himself
- Ability to create new ARs (in which their state is set to Open)
- Ability to set Testing ARs to Open (implying unsatisfactory resolution) or to Resolved (implying satisfactory resolution).
- Ability to modify AR information in particular states. (The particular information—i.e. fields—that may be modified depend on the state and will be agreed between MakaluMedia and ESA during the detailed design phase.)
- Received email-based notifications of relevant ARTS events. (The particular events for which notifications will be sent will be defined and agreed between MakaluMedia and ESA during the detailed design phase.)
- All AR and Action browsing and viewing services, as specified in Section 0.

### Actionees

The Actionees within an ARTS Project are the ARTS Users responsible for the completion of tasks (Actions) assigned to them. There may be multiple Actionees defined per Project.

Specifically, ARTS provides the Actionee with the following services:

- Ability to set In-Progress Actions to Responded.
- Ability to modify Action information in particular states. (The particular information—i.e. fields—that may be modified depend on the state and will be agreed between MakaluMedia and ESA during the detailed design phase.)
- Received email-based notifications of relevant ARTS events. (The particular events for which notifications will be sent will be defined and agreed between MakaluMedia and ESA during the detailed design phase.)
- All AR and Action browsing and viewing services, as specified in Section 0

### Guests

Guests are ARTS Users authorised to browse, without modification, the information recorded in an ARTS Project.

Specifically, ARTS provides the Guest with the following services:

- All AR and Action browsing and viewing services, as specified in Section 0.
External Actionees (Special Role)

A special role exists with the ARTS Project known as *External Actionee*. External Actionees are identical to Actionees, with the exception that they are only able to view Actions (and parent Anomaly Reports) assigned specifically to them. The AR and Action browsing and viewing services provided to External Actionees are defined in Section 0.

Data Access Services

ARTS provides to all Project Roles, the following AR and Action data access services:

- **Listing.** Generate Display Lists of ARs or Actions, based on pre-defined queries or user-defined queries.
- **Browsing.** Viewing of any AR or Action present in a Display List. When viewing an AR or Action, ARTS shall provide navigation services to the next or previous AR/Action in the list.
- **Exporting.** The ability to export as an emailed attachment, a single browsed or the complete results of a Display List of ARs or Actions. This Export Function is different than data export on the Communication Interface. ARs and Actions that are exported via the Export Function are encoded in tab-delimited Unix-formatted ASCII text files suitable for import into desktop applications such as EXCEL or ACCESS. ARTS supports a configurable limit to the data export filesize. Exceeding this limit results in an abort of the export, and a warning returned to the user.
- **Printing.** ARTS allows the user to access a printable-version of any browsed AR or Action.
- **Processing.** Depending on the viewer’s role and the state of AR or Action, the ability to directly enter a processing mode for ARs and Actions present in a Display List.

The particular headings and their sort preference (ascending/descending) for columnar lists of displayed ARs and Actions will be agreed between ESA and MakaluMedia during the detailed design phase.

9.1.1 Pre-Defined Queries

The generation of Display Lists of ARs or Actions within a Project are provided to ARTS Users (Supervisor, Deputy Supervisor, Originator, Actionee, External Actionee and Guest) based on the following pre-defined queries:
<table>
<thead>
<tr>
<th>ID</th>
<th>Query</th>
<th>S/D</th>
<th>O</th>
<th>A</th>
<th>G</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>All Ars</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>All ARs in the Open state</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>All non-closed ARs (i.e. Open, Pending, Testing or Resolved)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>All ARs created by the User</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>All non-closed ARs created by the User</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>All ARs containing non-closed Actions assigned to the User</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>All ARs with Actions Assigned to the User</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>8</td>
<td>All ARs created by the User presently in Testing state</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>All Actions</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>All non-closed Actions</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>All overdue Actions</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>All Actions assigned to the User</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>All non-closed Actions assigned to the User</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>All overdue Actions assigned to the User</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
</tbody>
</table>

9.1.2 User-Defined Queries

Display Lists of ARs or Actions within a Project may also be generated through custom search queries. Such queries may be made through a logical AND search on an arbitrary set of attributes. This includes all Role-viewable attributes of the entity, AR or Action.
10 Histories

In ARTS, a History is a logical element that stores information about other logical elements throughout their lifecycles.

The structure of a History is as follows:

- **History ID.** A non-modifiable, automatically created numeric attribute that uniquely identifies the History within the Instance.

- **Parent Entity Type.** The type of logical entity to which the History is associated. The architecture of ARTS allows a History to be associated to an arbitrary entity (e.g. AR, Action, Project, User or the Instance itself), the first release of ARTS implements History associations to Anomaly Reports and Actions.

- **Parent Entity ID.** The unique identifier of the Parent Entity.

- **History.** The History is a free text of arbitrary size that stores information related to lifecycle of its Parent Entity.

Histories supports two kinds of entries, described by example as follows with respect to ARs and Actions:

- **Automatic Entries.** Each time that an AR or an Action is modified, an associated time stamped entry is automatically recorded in the History, containing information related to the change (e.g. the user that made the change, the states prior to and following the change etc.)

- **Manual Entries.** Depending on the state of an AR or Action, the user may append the History with a manual entry. This would be used to add additional notes to the AR or Action, etc.

To provide maximum flexibility, the Supervisor, depending on the state of the AR or Action, may modify the complete History contents. Consider an Action that has experienced a complex lifecycle, including several clarifications and resolutions of misunderstandings, etc. When the Action is finally closed, the Supervisor may wish to modify the History, removing all entries that are not useful for post analysis and review of the Action.

11 Relations

In ARTS, a Relation is a logical element that relates two logical entities, for the purpose of this section called Entity A and Entity B. In the case of the first release of ARTS, a Relation represents the relationship between two Anomaly Reports.

The structure of a Relation is as follows:
• **Relation ID.** A non-modifiable, automatically created numeric attribute that uniquely identifies the Relation within the Instance.

• **Entity A Type.** The type of logical entity corresponding to one half of the Relation pair. The architecture of ARTS allows a Relation to be established between any arbitrary entities (e.g. AR, Action, Project, User or the Instance itself), however, the first release of ARTS implements Relations only between Anomaly Reports.

• **Entity A ID.** The unique identifier of Entity A.

• **Entity B Type.** The type of logical entity corresponding to the second half of the Relation pair.

• **Entity B ID.** The unique identifier of Entity A.

Relations are established and deleted through the management services (provided to the appropriate Role, depending on the context) associated with the related entities, e.g. Anomaly Reports.

## 12 Notifications

ARTS informs users of important events in the lifecycle of Anomaly Reports and Actions via email notifications. There are two classes of notifications supported by ARTS:

• **Internal Notifications** are sent to ARTS users when ARs and Actions are modified, or when certain system events occur. (The particular events for which notifications are sent will be defined in the detailed design phase.)

• **External Notifications** are sent on the discretion of the Supervisor to Notification Lists (possibly containing non-ARTS users) that are associated to second-level nodes in the AR Classification Hierarchy (e.g. Segment/Element).

ARTS does not provide services for the management of Notification Lists, but rather they are maintained manually by the Administrator or Owner as addresses stored in ASCII text files on the host server.

Notifications (External and Internal) take the form of an ASCII text email, and always contain the following information, *in addition to the text specifically relevant to the notification*:

• A Subject line relevant to the notification and prefixed with the text [ARTS]

• An introductory text indicating to the reader that the enclosed notification comes from the ARTS system

• Where applicable, URL(s) leading directly into the ARTS system to the relevant ARs and/or Actions

• An indication of all copied parties on the notification
A footer common to all notifications with general administrative information: (1) the ARTS Instance unique identifier, (2) the relevant Project identifier, (3) contact name and email for the relevant Project Supervisor in case of questions.

A FROM address corresponding to the Project Supervisor (so that any replies will be directed to him/her)
13 Attachments

ARTS provides the support of file attachments associated to Anomaly Reports and Actions.

- Files may be uploaded, one at a time, and associated with any AR or Action. The maximum number of files that may be associated with any AR or Action is determined by configuration.
- Files associated to ARs and Actions may be created, replaced and deleted.
- ARTS imposes a configurable limit to the maximum size of files that may be uploaded.
- Files are stored in the filesystem on the host platform.
- Files are not included in data exports, since the same users that perform an export have direct download access to these attachments from the ARTS User Interface.
- Files are included with Actions delegated to Delegate systems.
- Deletion of Actions and ARs results in the deletion of any associated files.
14 Communication Interface

Overview

A key aspect of ARTS is the ability to include Delegates in the lifecycle of Actions. A Delegate may be another Project within the ARTS Instance, or a Project within an external Foreign System. The assignment and processing of actions between an ARTS Project and a Delegate is based on ACI Messages (ARTS Communication Interface Messages) exchanged on its Communication Interface (CI). The mechanism by which messages are physically transmitted is electronic mail.

ACI Messages generated within ARTS may be sent to known entities registered within the Instance’s Systems Registry (described in Section 0), or on initiation of the Administrator to a specified email address.

Although the transmission and reception of CI messages generates notifications to the relevant ARTS Supervisors and/or Administrator, the Communication Interface’s protocol itself does not support message confirmation. This limitation implies the need for human intervention in the case that transmitted messages are not received, or are corrupted in transmission. On the other hand, the simplicity and ubiquity of the messaging technology (electronic mail) allows ARTS to communicate with other systems in a manner compatible with most networks, firewalls, etc.

ACI Messages

14.1.1 Message Structure

An ACI Message (ARTS Communication Interface Message) takes the form of an RFC-2822 compliant ASCII-encoded email message, with the following structure:

- **Return-Path.** The return-path field of the message contains the email address of the Administrator of the originating system, so that he will received undelivered bounced messages.
- **From.** The from field of the message may contains either the email address of the Administrator of the originating system, or the email address of the Supervisor, depending on the context (to be further defined in the detailed design phase).
- **Subject.** The subject field contains the prefix string “[ARTS CI] [message-ID]” where message-ID is the identifier of the message taken from the Administrative Header (described below) content.
- **Message-Id.** This field contains the unique message Id.
- **Arts-Ci-Command.** This field contains the Arts CI Command.
Project-Id. This contains the recipient project’s Id. (required by messages Action Update, Action Respond, Action Reject, Action Complete & Action Update Reject)

Body. The body of the email contains the message contents encoded in an XML-like structure. The message contents comprise two primary components: (1) an Administrative Header, and (2) a Content Block.

14.1.2 Administrative Header

The Administrative Header component of an ARTS CI messages contains the following:

- Message ID (Unique Id))
- Transaction ID (required by messages Pair Request, Pair Request Confirm & Pair Request Reject)
- Time Sent
- ARTS CI Command
- Sender System ID
- Sender system Short Name
- Sender system CI Email Address
- Sender system Administrator Email Address
- Sender Email Address
- Sender Project ID (required in messages Action Update, Action Reject, Action Complete, Action Respond, Action Update Reject, Pair Request, Pair Request Confirm, Pair Request Reject & Pair Remove)
- Sender Project Shortname (required in messages Action Update, Action Reject, Action Complete, Action Respond, Action Update Reject, Pair Request, Pair Request Confirm & Pair Remove)
- Sender Project Longname (required in messages Action Update, Action Reject, Action Complete, Action Respond, Action Update Reject, Pair Request, Pair Request Confirm & Pair Remove)
- Foreign Project Id (required in messages Action Update, Action Reject, Action Complete, Action Respond, Action Update Reject, Pair Request, Pair Request Confirm, Pair Request Reject & Pair Remove)
- Sender Action Id (required in messages Action Update, Action Reject, Action Complete, Action Respond)
• Foreign Action Id (required in messages Action Respond, Action Update Reject)
• Pair Relation (only Pair Remove Message)
• Sender Supervisor Email Address (optional)

Services
The following services are defined on the Communication Interface:

• Action Update. Sent from an originating system to a Delegate requesting the Delegate Supervisor to allow the creation or update of a Delegate Action on the Delegate system.

• Action Update Reject. Sent from a Delegate system to an originating system rejecting an Action Update (Delegated Action) when it is the first Action Update for that Action.

• Action Respond. Sent from a Delegate system to an originating system requesting to accept a response to a delegated Action.

• Action Reject. Sent from an originating system to a Delegate requesting the Delegate Supervisor to allow the rejection (set state to Rejected) of an existing Delegate Action.

• Action Complete. Sent from an originating system to a Delegate requesting the Delegate Supervisor to allow the completion (set state to Completed) of an existing Delegate Action.

• Registry Update. Sent from an originating system to a destination system, requesting the destination system to create or update a record of the originating system’s identifying information and Delegates.

• Pair Request. Request from an originating system to a destination system (destination can be the same system as the originator’s system), requesting that a project pairing can be made between a project on the originator’s system and a project on the destination system.

• Pair Request Confirm. Sent from the destination system to the originating system to confirm a project pairing can be set up.

• Pair Request Reject. Sent from the destination system to the originating system to reject requested project pairing.

• Pair Request Remove. Sent from either the originating system or the destination system to remove a project pairing.

Infrastructure
Since the mechanism for exchange of ACI Messages is electronic mail, each ARTS instance therefore requires access to a dedicated and operational POP3 email account to which incoming
messages are received, and an operational SMTP mail server through which outgoing messages are sent. (See Section 16 for more details about the application environment requirements.)

15 User Interface

ARTS is a web-based application, meaning that its primary tool for viewing the user interface is a web browser—e.g. Netscape or Internet Explorer. As such, a number of considerations and constraints must be taken into account in the development of the application, as described in this section of the document.

Browser Considerations

This section of the document discusses a number of considerations introduced by the fact that the ARTS user interface shall be accessed via a web browser.

15.1.1 Compatibility

The particular display of a web page is completely dependent on the rendering engine of the particular web browser, and can differ between browser make (e.g. Netscape and Internet Explorer), between versions of the same browser (e.g. Netscape 3 and Netscape 4) and even between the same browser version running on different platforms (e.g. Internet Explorer 5 for Windows, and Internet Explorer 5 for MacOS.) For this reason, it is necessary in any web-development project to specify the target browser intended for the project.

While MakaluMedia’s development philosophy targets maximum browser compatibility, the particular browser for which ARTS will be guaranteed compatible is **Microsoft Internet Explorer, version 5, for Windows OS**. Javascript must also be enabled on Internet Explorer for ARTS to function correctly.

15.1.2 Client-Side Technology

The display of web pages by browsers was originally based on the rendering of a document encoded in the HTML structural mark-up language. For various reasons, the past years have seen the introduction of a number of complements to HTML, including JavaScript, Dynamic HTML (DHTML), Cascading Style Sheets (CSS), etc. These technologies, known as client-side technologies, depend on the web browser implementation for correct display or function. Whereas the differences in the display of HTML between browsers today are minimal, the implementation of new client-side technologies can vary greatly, even between minor releases of the same browser on the same platform.

In order to minimise project risks, and to promote cross-browser compatibility, the general approach in the ARTS project shall be minimal reliance on client-side technologies.
15.1.3 Layout & Display Techniques

The layout of any website or application should take into account the expected monitor size of the intended viewer. It is generally accepted today that a reasonable target resolution is 800 x 600 pixels, and this is the assumption on which ARTS shall be developed. Furthermore, this is in line with most other official ESA websites.

The definition of styles—i.e. whether to use CSS or HTML font tags—shall be left open until the implementation phase of the project. Independent of the style technology chosen, the ARTS user interface will provide a control to increase or decrease the default font size, on which all other font sizes are relatively specified.

15.1.4 Graphics Technologies

The large majority of graphics (images) viewable on the web today are of the format GIF or JPEG. In general, the GIF format (which uses a lossless compression technology) is used for computer-generated graphics, and the JPEG format (which uses a lossy compression technology) is used for natural images.

The Internet community has devised an open graphic format known as PNG. Although still without fully consistent rendering support across browsers and platforms, the format is growing in popularity. Due to patent copyright issues, graphic generation technologies (such as the open source solutions available to PHP) generate PNG graphics, instead of GIF.

Graphics developed for ARTS external to the application (but displayed within the ARTS User Interface) shall be of the GIF, JPEG or PNG format. Graphics developed dynamically by ARTS shall be of the PNG format.

Usability Consideration

A good user interface is one in which the user can quickly and efficiently achieve his intended task. There are a wide number of techniques known, and intended for use in ARTS, to achieve a high degree of usability:

- Minimize the number of grouped decision elements
- Maximize the ratio of information-bearing pixels to non-information-bearing pixels
- Implement pages that load quickly in the browser
- Provision of feedback to user actions
- Provide a navigation scheme that prevents the user from losing his bearing in the website structure
Visual Design

A good visual design relates to a good usability design, but further extends into the topics of layout, dimensions, colours, fonts, corporate identity, etc. The ARTS development will consider the following:

- A fixed-width layout in order to control text width to known readable lengths.
- The use of no more than two font faces on a page.
- The use of fonts and colours that are consistent with ESA’s Corporate websites.
- The use of consistent alignment of elements on the page.
- The use of contrast to highlight the priority of elements on the page.
- The use of proximity, i.e. the grouping of related items.

User Interface Considerations

The ARTS User Interface, to be developed within this project, shall consider the following:

- User-modifiable elements shall be presented as form elements, while non-modifiable elements shall be presented as screen text.
- ARTS shall validate all submitted user information, for both completeness and syntax correctness.
- Submitted forms containing incomplete or invalid data shall be returned to the user with a warning, highlighting (where applicable) the erroneous fields in red, and pre-filling the form with the previously submitted data.
- To the greatest extent possible, forms shall be presented to the user pre-filled with derived information.
- Where data format is critical (e.g. dates and times), ARTS shall present to the user the appropriate format, and shall validate the submitted data.

Online Help

The ARTS User Interface shall provide help documentation to the user based on the context of the present use. Help shall be indicated to the user through consistently used small icons throughout the User Interface. When clicking on a Help icon, a new window shall be created, containing the help text.
Help Text are associated to the ARTS Instance. ARTS shall provide no services for the management of Help Text items. The particular locations within the User Interface where Help Text is located shall be agreed between ESA and MakaluMedia during the detailed design phase.
16 Application Environment

This section of the document describes the considerations surrounding the environment in which the ARTS application will be operated.

Platform Infrastructure

The ARTS application development will be based to the greatest extent possible on open-source technologies, and where possible using products that exist on multiple platforms.

- **Operating System.** The ARTS application will be developed under the Linux operating system, and will be operated under ESA’s standard Solaris 2.5.6

- **Web Server.** The web server that shall be used to access ARTS is Apache, version 1.3.23. Each installation of ARTS of the same machine, should be configured as a separate Apache Virtual Host, and bound to a unique address, e.g. ifms.arts.esa.int or nctrs.arts.esa.int, etc. Should many ARTS installations be supported on the same machine, the web server should be configured by an administrator knowledgeable in the performance tuning of Apache, in particular the appropriate settings of MaxSpareServers, MaxClients and MaxRequestsPerChild.

- **Web Application Language.** The application language used to develop ARTS is PHP, version 4.1.1.

- **Support Scripting Language.** The scripting language used for the development of support tools and applications is Perl, version 5.6.

- **Database.** The database intended for ARTS data storage is MySQL, version 3.23.39.

- **Incoming Email Support.** In order to receive incoming emails, each ARTS installation requires access to a dedicated email account (not necessarily local) that is not restricted in disk space, and is accessible via the POP3 protocol. A suggested approach would be to define the email accounts in a consistent manner, e.g. ifms@arts.esa.int and nctrs@arts.esa.int, etc.

- **Outgoing Email Support.** In order to send outgoing emails, ARTS requires access to either a local SMTP server (e.g. sendmail, running under Linux/Solaris) or an external SMTP server. It is also necessary that the SMTP server does not modify the email headers as specified by ARTS.

Server Infrastructure

Given the dimensioning of most server machines available today, it is unlikely that server resources would become a limiting factor in the operation of ARTS. Many ARTS installation (e.g. 100
projects, with a total of 500,000 database records) could reliably be supported on a Linux-based machine with a 1.5 GHz processor, 512 MB system memory and a 50 GB hard drive.

The ARTS General Concept document discusses the access of ARTS when behind a firewall. It should be noted that considerations related to such access are beyond the scope of the ARTS application development.

### Application Infrastructure

In order to minimize development time and risks, the ARTS web application shall use as a foundation the MakaluMedia Application Infrastructure Modules (AIM). The AIM is a web-application framework (toolkit) under continual development and enhancement by MakaluMedia that provides general application infrastructure support, such as:

- Database access abstraction.
- Session storage and management
- User authentication and management
- Environment access
- Logging
- Markup syntax support

The AIM has been well tested in its support of operational websites and applications during the past four years (with respect to the original date of this document).

### Logging

The ARTS application maintains a file on the host platform in which system events and information are logged. Three levels of logged events are supported:

- **L1.** System events and information that are of sufficient importance that an email notification is sent to the Instance *Administrator*.

- **L2.** System events and information that should be logged for the purpose of troubleshooting, but do not require Administrator notification.

- **L3.** System events and information that should only be logged for the purpose of debugging. Only when the system, through configuration, is set to *debug mode*, shall L3 events be logged.

L1 events are identified in the documentation associated to the design phase. L2 and L3 events are only documented in the application source code.

In order to prevent excessive use of disk space, ARTS rotates the log file when it exceeds a configurable size. The rotated log file overwrites any existing rotated log file, and therefore the
maximum disk space occupied by the active and rotated log files is twice the configured maximum log file size.

**Data Backup & Restoration**

The complete data state of an ARTS installation at any point in time is comprised of (1) the application’s data stored in the MySQL database, and (2) the application’s data stored in the host platform’s file system. (It is *not* intended to store binary data in the MySQL database.) In order to perform a successful restoration, both of these components would need to be included in a well-considered data backup programme.

In order to support the first component (the backup of MySQL), the ARTS application will include a transparent service that exports on a daily basis the data stored in the MySQL database to ASCII text files stored on the host platform. In addition, the application will provide services to the Administrator to manage these data snapshots, deleting a range of old snapshots, and restoring any particular snapshot to the database.

In order to support the second component (the backup of the host platform’s file system contents), the owner or administrator of the ARTS host platform is responsible for providing a reliable daily backup service. Such a service is clearly outside the scope of the ARTS application software.
17 User Requirements

The following sections contain the User Requirements that form the baseline for the ARTS development.

General

1 ARTS shall support concurrent access, from the same or different users, without data loss.

2 To the extent possible, automated scripts shall support the installation of an ARTS Instance.

Systems & Delegates

3 All systems to which Actions may be delegated shall be registered in the ARTS Systems Registry.

4 The Systems Registry shall contain at least one entry, corresponding to the Default System. The Default System shall be initialised during ARTS installation.

5 All Systems in the Systems Registry shall contain the identifying information as specified in Section 0.

6 All Delegates to which Actions may be delegated shall be registered in the ARTS Systems Registry.

7 All Delegates in the Systems Registry shall contain the identifying information as specified in Section 0.

8 ARTS shall provide the Administrator with the services to manage the Systems Registry as specified in Section 0.

9 A Foreign System may not be deleted from the Systems Registry if there are associated delegated actions in the Remote In-Progress or the Remote Responded state.

10 ARTS shall provide the Administrator with the service to announce or update his Instance and Delegates to known systems existing in the Systems Registry, or to an email address manually specified.

11 ARTS shall allow the Administrator, when announcing or updating the availability of his Instance, to select zero or more Projects to nominate as available Delegates.

12 ARTS shall not provide services to ensure the consistency at any given time between information about an entry in the Systems Registry and the actual current state of the Foreign System.
Users

17.1.1 General
13 ARTS Users shall be defined once per ARTS Instance, and stored in the ARTS User Registry.
14 ARTS Users shall be described by the attributes defined in Section 0.
15 ARTS shall provide all service to the Administrator necessary to manage Users as described in Section 0.
16 ARTS Users shall be provided the service to modify their own administrative information (name, email address and password.)

17.1.2 Administrator
17 The Administrator shall be a regular ARTS User identified through configuration to be the Administrator.
18 ARTS shall initialise the creation of the first User, the Administrator, during installation.
19 ARTS shall not allow the deletion of the User identified as the Administrator.
20 Upon login, ARTS shall inform the Administrator of any pending ACI Messages.
21 ARTS shall provide the service to the Administrator to export the entire Instance database, and transmit the files as a compressed, Unix tar archive to the Administrator’s email address. The format of the files is that of the ARTS database backup snapshots.
22 With respect to AR and Action lifecycles, the Administrator shall be a superuser, meaning that ARTS shall provide him, at any time, with all services available to Supervisors, Originators and Actions in order to manage ARs and Actions as if they were his own.

17.1.3 Login & Authentication
23 ARTS shall only be accessible by registered Users, through login with a valid username and password.
24 Invalid logins shall return the login form, pre-filled with any submitted data, and shall prompt a warning to the user without indicating specifically which login field (username or password) was invalid.
25 The ARTS login screen shall provide the user with the service to send his password to his registered email address.
26 The ARTS login screen shall provide an introduction to the system, and the possibility to download reference documentation (although ARTS will not provide services to manage such documentation.)
ARTS shall provide a service to unregistered Users to communicate with the Administrator via a form-based email.

ARTS shall provide at all times the service to return to the Login Screen by logging out of the application.

Upon Login, ARTS shall present the User with a Welcome Page listing links to all Projects (if any) to which he is associated, and the name of the associated Supervisors. The Supervisor’s name is a link to his profile.

At any time a User may return to the Welcome Page (without logging out), thereby allowing within the same session access to multiple projects (and therefore possibly multiple Roles).

17.1.4 User Profile & Communication

ARTS Users identified in the User Interface shall be identified as links to the corresponding Profile Page.

ARTS shall allow a User at any time to access his own Profile Page.

ARTS User Profile Pages display the information and provide the services as specified in Section 0.

Projects

An ARTS Project shall be defined according to the attributes specified in Section 0.

ARTS shall provide the Administrator with all Project-related services specified in Section 0.

ARTS supports Project-related configuration elements as defined in Section 0.

ARTS shall check for the presence of all required configuration elements, and shall report to the Administrator a warning in the case of problems.

ARTS shall validate the syntax of all Project configuration elements, and shall report to the Administrator a warning in the case of problems.

Anomaly Reports

ARTS shall support Anomaly Reports (AR) with a structure as specified in Section 0.

Each Anomaly Report shall have an associated History.

An AR shall be uniquely identifiable within an ARTS Instance.

ARTS shall allow ARs to assume the states as specified in Section 0.

ARTS shall support an AR lifecycle as specified in Section 0.
ARTS shall provide all AR management services specified in Section 0.

When an Action is delegated to a Delegate system, the parent AR shall be transmitted along with the Action to the Delegate system, and shall result in a Foreign Anomaly Report on the Delegate system.

ARTS shall allow the association between two ARs within the same Project. Associations are displayed to the User as links when browsing the ARs.

ARTS shall allow ARs to be classified in three levels, known as the Project’s Classification Hierarchy, to be defined on a per-Project basis by the Administrator.

ARTS shall only present to the User valid choices for an AR’s Classification Hierarchy.

The Classification Hierarchy shall be defined on a per-Project basis, in an XML file (one file per project) located on the host machine’s file system, with contents as specified in Section 0.

ARTS shall provide no services for the management of the files containing the Classification Hierarchy, but shall validate their syntax and contents. Errors shall be reported to the Administrator in the Projects Registry.

**Actions**

ARTS shall support the management of two type of Actions, Local and Foreign, as specified in Section 0.

ARTS shall support Local Actions with a structure as specified in Section 8.1.2.

ARTS shall allow Local Actions to assume the states defined in Section 8.1.5.

ARTS shall support a lifecycle for Local Actions as specified in Section 8.1.7.

ARTS shall support Delegate Actions with a structure as specified in Section 8.1.3.

ARTS shall allow Delegate Actions to assume the states defined in Section 8.1.6.

ARTS shall support a lifecycle for Delegate Actions as specified in Section 8.1.8.

Each Action shall have an associated History.

An Action shall be uniquely identifiable within an ARTS Instance.

**Roles & Services**

**17.1.5 General**

ARTS shall support four Project-related permission levels, also known as Roles—Supervisor, Originator, Actionee and Guest.
ARTS Project Roles are hierarchical, in that a particular Role inherits the services available to Roles lower in the hierarchy.

17.1.6 Supervisor

Upon login, ARTS shall inform the Supervisor of any pending ACI Messages.

ARTS shall provide the Supervisor with all services described in Section 0.

17.1.7 Deputy Supervisor

ARTS shall provide the Deputy Supervisor with all services described in Section Error! Reference source not found..

17.1.8 Originator

ARTS shall provide the Originator with all services described in Section 0.

17.1.9 Actionee

ARTS shall provide the Actionee with all services described in Section 0.

17.1.10 Guest

ARTS shall provide the Guest with all services described in Section 0.

17.1.11 External Actionee

ARTS shall provide the External Actionee with all services described in Section 0.

17.1.12 Data Access Services

ARTS shall provide Project-related Users with the services defined in Section 0.

ARTS shall provide Users with the service of creating Display Lists from pre-defined queries as mentioned in Section 9.1.1.

ARTS shall provide Users with the service of creating Display Lists from user-defined queries as mentioned in Section 9.1.2.

ARTS shall provide all Users the service to export a browsed AR or Action.

ARTS shall provide all Users the service to export all ARs or Actions found within a Display List.

ARTS shall transmit exported data via an email attachment to the requesting User’s registered email address.
The exported data shall be contained in a tab-delimited, Unix-formatted, ASCII text file, with one record per line, suitable for import into desktop applications such as EXCEL.

ARTS shall impose a configurable limit to the file size of data exports. Exports exceeding this limit are aborted and a warning is displayed to the user.

ARTS shall provide within the User Interface access to a printable version of any viewed AR or Action.

**Histories**

ARTS shall support a logical element known as a History, with a structure as defined in Section 10.

ARTS shall associate a History with each AR and Action.

ARTS shall support both automatic and manual entries to be made in a History.

A User with Supervisor level permission (or the Administrator) shall be provided with the service (depending on the AR or Action state) to fully modify the History contents.

Each time that an AR or Action is modified, ARTS shall record, the date, time, user and nature of the change in the History.

**Notifications**

ARTS shall support the transmission of information via email to Notification Lists associated with second-level nodes in the Classification Hierarchy. Notifications to such lists are known as *External Notifications*.

Notification Lists shall be defined in Unix-formatted, ASCII text files, with one email address per line, and located on the host server’s file system.

The association of Notification Lists to Classification Hierarchy nodes shall be based on the contents of the Notification List filename.

ARTS shall not provide services to manage Notification Lists.

ARTS shall only make Notification services available on ARs that have been classified to a Classification Hierarchy node for which a Notification List is present.

Each time that an External Notification is initiated, ARTS shall make a record of that notification in the associated AR’s History.

ARTS shall support the transmission of emails to Users based on certain system events. Such notifications are known as *Internal Notifications*. (The particular events to which Internal Notifications will be sent shall be agreed between ESA and MakaluMedia during the detailed design phase.)
Attachments

90 ARTS shall support the association of files with Anomaly Reports and Actions as specified in Section 13.

Communication Interface

91 ARTS shall support a Communication Interface (CI) based on the exchange of messages (known as ACI Messages) taking the form of RFC-2822 compliant email messages.

92 The data contained in ACI Messages shall be encapsulated in XML data structures.

93 The behaviour of the CI shall be independent of the managing application in terms of software, operating system and network.

94 The scope of the CI shall be limited to the exchanged data, without access to the other information stored on either application.

95 In the context of Customer and Service Instances, new problems affecting the context of the two systems can only be generated at the Customer side.

96 ARTS shall receive ACI Messages in an email account provided by the Application Environment, and dedicated to the ARTS Instance.

97 ARTS shall transmit ACI Messages using an SMTP server provided by the Application Environment.

98 The ARTS CI shall support the services specified in Section 0.

99 The structure of an ACI Message shall correspond to that specified in Section 0.

100 ARTS shall not confirm receipt of any transmitted ACI Message.

101 Upon transmission of an ACI Message, ARTS shall send a notification email to the responsible person (depending on the ACI Message, an Administrator or Supervisor) with a copy to the originator.

102 ARTS shall require the approval of (depending on the context) a Supervisor or Administrator, before executing the action (or service) requested in a received ACI Message.

103 ARTS shall validate the recipient and the message syntax of ACI Messages.

104 Aborts based on an invalid recipient or message syntax shall generate an email notification to the relevant parties (Supervisor/Administrator depending on context) on both sides of the transaction, and display a warning within the User Interface.
User Interface

17.1.13 Browser Considerations

105 The ARTS User Interface shall be accessible via a web browser.

106 The ARTS User Interface shall be compatible with the Microsoft Internet Explorer web browser, version 5, running on the Microsoft Windows98 platform, with JavaScript and Java enabled.

17.1.14 Client-Side Technology

107 ARTS shall rely on the minimum extent possible on client-side technologies (e.g. JavaScript, DHTML, CSS, etc.)

17.1.15 Layout & Display Techniques

108 The ARTS User Interface shall occupy a width suitable for display in a monitor with resolution of 800 x 600 pixels.

109 ARTS shall impose no constraints on vertical layout.

110 HTML text displayed in ARTS shall be based on a default font size, which can be modified by the User, based on a pre-defined selection, to suit their individual taste. The User-specified default font size is persistent throughout a session.

17.1.16 Graphic Technologies

111 Graphics within the ARTS User Interface developed external to the application shall be of the GIF, JPEG or PNG format.

112 Graphics generated dynamically by the ARTS application shall be of the PNG format.

17.1.17 Visual Design

113 The ARTS User Interface will be presented within a fixed width on the screen.

114 The fonts and colours used in ARTS will be consistent with ESA’s corporate websites.

17.1.18 User Interface Considerations

115 All User Interface considerations as listed in Section 0 shall be implemented.

17.1.19 Online Help

116 The ARTS User Interface shall support the presentation of Help icons leading to context-relevant help text.
ARTS User Requirements Document
MM-ESA-ARTS-URD • Issue 3.0 • 2004-09-30 • Page 55

117 ARTS shall provide no services for the definition or management of help text.

Application Environment

17.1.20 Platform Infrastructure

118 ARTS shall run operationally under the Solaris 2.6 operating system.

119 ARTS shall not restrict the number of installations that could be supported on the same host server.

120 The ARTS User Interface shall be serviced by the Apache web server, version 1.3.23, including the PHP script interpreter module, version 4.1.1.

121 The development language of the ARTS primary web application shall be PHP.

122 The scripting language used for the development of support applications and utilities shall be Perl, version 5.6.

123 The database management system supporting the ARTS application shall be MySQL, version 3.23.39.

124 Incoming email for the ARTS application shall be stored in an email account accessible via the POP3 protocol. Each ARTS installation shall have available a single dedicated POP email account.

125 Outgoing email from the ARTS application shall be sent through an SMTP server that is configured to not modify email headers.

17.1.21 Logging

126 ARTS shall maintain an active log file on the host server’s file system.

127 ARTS shall support three levels of logging, L1 (in which logged events generate email notifications to the Administrator), L2 (in which no email notification is generated), and L3 (only enabled when the system is set to debug mode).

128 ARTS shall rotate the active log file when it exceeds a configurable size.

17.1.22 Data Backup & Restoration

129 ARTS shall support the daily or Administrator on-demand backup of the database table contents to ASCII text files on the host server platform. A set of table backups is known as a snapshot.

130 ARTS shall provide the Administrator with on-demand restoration of any available snapshot.
ARTS shall provide the Administrator with on-demand deletion of any specified snapshots or a range of snapshots specified by date.
CCN4-R2   Control of Email Notifications

A new section will be added to the page "Administration / Project Registry / Project Configuration" in order to allow the administrator to enable or disable the triggering of the notification mails for each triggering event and user role. The information on the settings will be stored into a new table.

CCN4-R5   System Suspension

Functionality shall be offered to the Administrator to suspend the system to all other users than the administrator. When ARTS is in this state:

The Administrator may continue to work with the system as usual

Users attempting to access the system are shown a screen stating: "Maintenance is Presently On-Going. Application Temporarily Inaccessible."

A red button is shown persistently in the header that reads "ARTS Suspended", to prevent the Administrator from forgetting that the system is in suspension.

When the Administrator places the system in suspension mode, all current login sessions will be cleanly closed.

A new section (tab) within the Administration interface will be created called "Activity". The services to "Suspend" and "Activate" ARTS will be found in this area.

The Activity section will additionally allow the Administrator to specify the planned date and time of the next scheduled outage, with the following behaviour:

If this field contains a valid date/time in the future, then the return screen from all users actions within ARTS shall contain a notification that ARTS is scheduled for suspension at the specified date time, and will provide an indication of how many minutes (from present) remains before the system goes into suspension.

If this field contains a valid date/time in the past, the system will do nothing.

The system will NOT automatically go into suspension at the specified date/time. In other words, this is not a feature to schedule a suspension, but rather only a feature to announce actively a suspension. (The admin will therefore always have to manually suspend the system at the time he desires.)
CCN4-R6

A new section (tab) within the Administration interface will be created called "Communication". This tab will contain an email form from which the Administrator can generate an email to be sent to all currently registered users. This service can be used to allow the Administrator to generate an email announcement to all ARTS users informing them of an upcoming maintenance outage, for example.

CCN4-R7 User Inactive Flag

An "Activity" flag will be added to each user in the form "Administration / Project Registry / Roles", the value of which will determine whether the user should appear as available for selection within drop-down user lists within the interface:

- Actionee assignment lists within Projects
- Role assignment/creation within the Project Registry

The default value for new users will be 1 (Active). When Active, a user's name will appear in selection lists within the interface. When the value is 0 (Inactive), the user's name will not appear in selection lists within the user interface.

The value of this flag will otherwise, however, not affect the user's ability to login and participate in ARTS.

CCN4-R8 New Action & New Report Services

Within the Action Details and Anomaly Report Details tabs, and sharing the area with the "Previous" and "Next" buttons, a new button will be included with the title "Create New Action". This button allows creating a new action associated to the current action's parent AR. (This button will not be shown on the Action Details screen when viewing Foreign Actions, since their parent AR is a different type of AR entity which is unmodifiable.)

On the Anomaly Report Details tab, a service will be provided "Create New Report" that operates in a similar way (but without having any relation to the currently viewed report.)

CCN4-R9 Display of Logged-In Users

A new section (tab) will be created in the Administration area called "Activity". This area will contain a listing of all currently logged in user sessions. The following information will be displayed in a sortable table:

- User full name (presented as a link to their area in the User Registry).
CCN4-R10 Report Closure Field

A new field, Report Closure, shall be added to the Report and Action entity, and shall record the date of Closure or Rejection of an Report and the date of Rejection or completion of an Action. Its content will be displayed when the Report is in Closed or Rejected state. The contents of this field shall be erased (but the history shall be updated consistently) when an Report is re-opened and the field will not be shown in the user interface.

This will affect display, print, export and PDF versions of Report and Action. This new field will be presented in the first "block" of the Report layout.

CCN4-R11 Short Access URLS

ARTS shall support short-access URLS of the following forms:

1. http://arts/?r=8792873492734928 (Action, using the UUID)
2. http://arts/?a=8792873492734928 (Action, using the UUID)
3. http://arts/?a=ARTS-123 (Action, using the constructed id PROJECT-ArID)
4. http://arts/?r=ARTS-123 (Report, using the constructed id PROJECT-ActionID)

Requests such as this would be handled as follows:

1) ARTS will determine whether the user is logged in.
2) If not, the login screen is shown.
3) If the user is logged in, or when he successfully logs in, ARTS determines the relevant Project, AR or Action within that project, and takes the user directly there.
4) If the ID is not found, the user is brought to the ARTS Home page.

CCN4-R14 ACI Proxy Report Creation

The ARTS communication interface shall be extended to allow paired systems to act as "proxy originators" in remote projects. In this way, a foreign system can, based on an existing local
Anomaly Report, raise an AR on a foreign system that is based on the data of the local AR. Once created on the foreign system, there is no logical connection between the ARs -- i.e. updates to the new AR are not notified to the originating system, and the Supervisor on the foreign system would be required to close the AR when in the Testing state.

In short: this feature exists simply to allow one ARTS system to raise an AR on another system, without participating in the lifecycle of the foreign created AR.

It will not be possible to send multiple ARs from a display list, due to the possibility of excessively sized email messages.

CCN4-R15

When foreign Reports (i.e. parent ARs of delegated Actions) are viewed by the Supervisor, a new service button shall be displayed entitled, "Raise Local AR". Selection of this service will create a new local Report (with the Supervisor as the original), using as far as possible the data in the foreign Report, and will pre-establish an Report relationship to the foreign Report on which the new local Report was based.

Once the local Report has been established, the "Raise Local Report " button shall change to "View Local Report " (to prevent creation of multiple local ARs based on the same foreign Report, and to allow quick access to the local Report from the foreign Report.)

There shall be a new project-level configuration switch introduced in the Administration area, to enable (or disable) the automatic creation of local Reports upon import of delegated Actions. This switch shall be in the "Disable" position by default.

The fields to be copied upon the raising of the local Report will be agreed with ESA during the work.

CCN4-R16 Report Lifecycle Modification

Modify the Report lifecycle so that Originators may modify an Report (Title, Description, Append History, etc.) while in Pending state.

CCN4-R17 Cross-Project Relations

The Manage Relationships facilities of a Report shall be modified to allow a Supervisor or the Originator to relate the present Report to a Report in another project (within the same AR installation). These shall be known as Foreign Relations. For this purpose the user interface shall provide a new field in the selection box to allow searching for Projects in which the user has a role (at least Guest).
In the user interface, the foreign relation will be presented as a link only in the case that the viewing user has a role in the foreign project (and is therefore authorised to view the related AR). Otherwise the foreign relation is presented as a plain text.

Updated screens include: Display, Print, PDF, Export and Download.

CCN4-R19 Advanced search for Actions

The advanced search for Actions shall be extended with the possibility to cross search also the parent's Report fields: originator and subsystem (at least). The selection box shall be updated and a new function shall be implemented to implement the search in the Report database.
CCN4-R13 Multiple Report Types

This requirement is by far the most complex requested for this CCN.

**Basic Requirement**

At present, ARTS is based on the definition of a single Report-type called “Anomaly Report”. It is requested by ESA to define a fixed number of report “types”, and then allow at project level to specify one of these defined types for use within the project. The Report-type for a given project is specified at creation time and can’t be changed afterwards.

The list of fixed type definitions (including labels) is provided by ESA in [R6].

Each defined Report Type includes a sub-set of the global set of defined AR fields in the MySQL database. Although the “labels” (that are shown in the User Interface) will vary per Report Type, the low-level fields will always map to the global set of AR fields in the database. This will allow to preserve consistency in the processing of the Reports. Furthermore, the type definitions (including labels) will be implemented at the database level, and not modifiable within the User Interface.

**Definition of the Classification within the User Interface**

Definition of the Classification structure will be provided as a service within the User Interface.

**Update to the Advance Search**

The Advanced search will be modified in order to load from the database the correct labels and data-types of the fields according to the Project’s Report-type.

**Miscellaneous Implications, Considerations & Clarifications**

Implementation of this requirement has a number of miscellaneous implications, considerations and clarifications that should be taken into account:

Implementation of multiple report types will require a number of architectural modifications and extensions, for example, to allow field definition mapping (to retain configurability), to allow mapping of field type to form element, to allow for new type-based sequencing (see below), etc.

It is better, both for flexibility and for implementation costs, not to hard-code the eight report-types and their fields into the ARTS database.

A configurable system will be implemented such that the number and type of reports can be easily extended. (And we stress that this approach is more cost-effective, especially in terms of future maintenance, than hard-coding the report types and fields.)
All report-types will have a standard set of attributes, which will be mandatory (but they can still be hidden and/or the label can be modified) for each report. The other attributes will be configurable. The mandatory fields are the status, the classification, the observation, the originator, the dates and few others (mainly those in the Header section).

For each of the report's fields (mandatory or not) the meta-data will define the group, the display properties (type of the field, multi-line, scrollbar, etc.), optional defaults etc.

There will be a set of entirely user-definable fields for each report-type. There will be also a set of user-definable "List-of-Values" to force particular values (like yes/no, criticality, urgency, etc.)

**Implementation Specifics**

The following bullets note a number of implementation details:

Changes of the DB Schema and Tables. We will introduce few tables to handle the meta-data describing the report-types and their fields.

Each report-type will be composed by a number of mandatory fields + additional fields.

The fields will be organized into "groups" with Title (for Display and PDF)

Each field (mandatory or additional) will be defined using meta-data, including type (text, number, date, etc), input layout (multi-line, scrollbars, etc), Visible (to hide un-used mandatory fields), etc.

The meta-data for the ESA defined report-type will be pre-loaded into the database.

We will not provide a user-MMI to modify the report-type meta-data (to reduce the cost)

Some field could be defined to be associated to a "List of values" (LOVs). Some LOVs will be pre-loaded (e.g. yes/no, On/Off, etc)

LOVs can be defined at 2 levels: (1) Project and (2) Report-type. (1) Each report will share the same LOVs. (2) LOVs are not shared between report-types.

We will provide a user-MMI to customize LOVs.

The fields which will be displayed in the Report list page will be customizable for each report-type (no MMI will be provided though).

The advanced search (at least the MMI) must be mostly re-written because we have to account for different report-type and their fields.

The functions "Previous" and "Next" has to be modified to conform with the new navigation.

The classification tree (CT) must be handled as follows:
- One CT must be defined at project level
- For a given report-type, if a private CT exists than this CT must be used
- otherwise the CT at the project level must be used

We will provide the population of the meta-data tables according to ESA report-type specifications.
We will provide a function to port the current DB data into the new DB schema.

Help documentation shall be updated to reflect the new Report Types.

ESA ACCEPT-123 Life-cycle

A selectable option at project level to enable or disable the automatic transition from Open to Pending shall be implemented. The modifications include a change in the DB schema, the implementation of the MMI section and the modification of the source to retrieve the project setting and behave consequently.