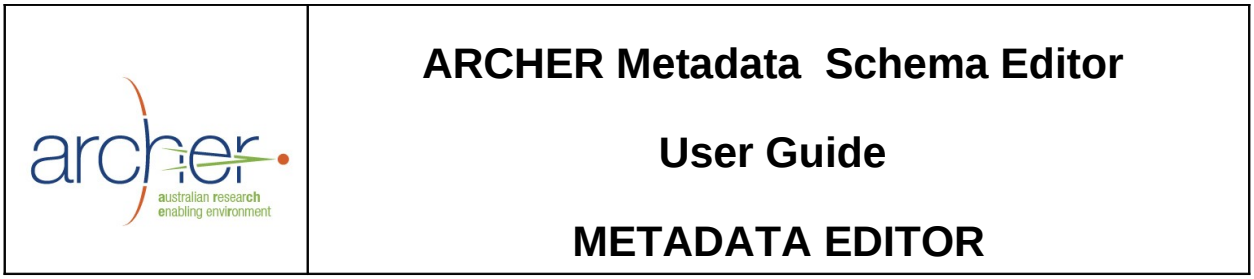

Version:	1.1
Date:	2008-08-26
Status:	Release

Change History

Version	*Date*	*Author*	*Description*
0.1D	2008-04-15	Ron Chernich	First Draft
1.0	2008-05-01	Ron Chernich	Revise and release
1.1CR	2008-08-22	Ron Chernich	Draft revision for new features
1.1	2008-08-26	Ron Chernich	Final release

Table of Contents

1. Introduction

1.1. Purpose of Document

This document provides basic how-to information for Users of the Metadata Schema Editor (MSE) that is used to create and edit metadata records.

1.2. Scope

The intended target audience for this document is end Users:

- It is assumed that the reader is familiar with basic metadata standards and metadata "good practice".
- It is assumed that the reader understands the problem domain for which metadata is to be maintained, or that they can ask a domain expert.
- It is expected that the reader has competence in the use of the common graphic user interface (GUI) "widgets".

1.3. Background

This MDE is designed to address the tasks of creating and maintaining high quality metadata based on specific, pre-defined metadata schemas. The purpose of a metadata schema is to specify what constitutes valid or meaningful metadata for a particular context. The goal is to allow metadata to be checked for completeness, consistency and (ideally) accuracy when it is saved, or at any time during the edit session as determined by the User.

By applying schema-based editing, the MDE is able to ensure that metadata records:

- Contain only elements valid for the underlying schema and schema version.
- Mandatory elements are present and any cardinality rules defined for the schema are applied.
- As far as is possible under the schema definition, element values are semantically correct.

However there are times when a User may decide that a record should be saved (made persistent), irrespective of any validation errors it may contain. The MDE accommodates this "user in charge" philosophy by delegating the responsibility for saving an invalid record to the underlying application which launched the editor. If this application permits the persisting of incomplete or invalid records, the editor will not stand in the way!

1.4. Environment

The MDE requires only a *"web browser"* and connection to a host service for operation. This may be provided by the Internet, an intranet, or even a local host service. The MDE has been tested and certified for operation under the following Internet browsers:

- Firefox 2.x (Microsoft Windows XP, Linux, and Mac OSX)
- Microsoft Internet Explorer 7.x

Operation in other environments is unsupported.

1.5. Context

The MDE must execute in the context of an Application that invokes the editor at appropriate times during the application's workflow and provides the means of loading and saving the metadata records to a persistent storage facility associated with the Application. This process will be transparent to the end User, making the editor appear to be part of the Application itself.

Each persisted record contains a reference to the metadata schema it complies to. Thus the schema also forms part of the record maintenance context, allowing the user to select and create additional elements as appropriate with high confidence in the result.

1.6. Conventions

The following typographical conventions are used in this Guide:

`Constant width`

Denotes user selectable menu options, or buttons.

`Constant width italic`

For replaceable parameters in dialog boxes or configuration files.

This icon indicates a warning or caution.

This icon indicates a tip, suggestion, or general rule.

1.7. Terms and Abbreviations

Application	a collection of tools which are used collectively to achieve a business process - roughly analogous to a program, as a normal user perceives it.
Metadata	in general: data describing data; in this context, the metadata consists of name-value pairs.
Metadata Schema	a formal specification of what is valid or meaningful metadata in a particular context.
User	a person with computer skills and sufficient domain knowledge to undertake creation and maintenance of metadata records.

Acronym	*Expansion*
MCP	Metadata Creation Package
JAR	Java Archive
JSON	JavaScript Object Notation
MDE	Metadata Editor
MDSR	Metadata Schema Repository
MMF	Metadata Management Facility
MSE	Metadata Schema Editor
MSF	Metadata Schema Facility
MSS	Metadata Schema Schema
RCP	(Eclipse) Rich Client Platform
UI	User Interface
URL	Universal Record Locator
XMI	XML Model Interchange
XML	eXtensible Mark-up Language
XSD	XML Schema Definition

Table 1

1.8. References

MCP-FRS *ARCHER Functional Specification – Metadata Creation Package,* Version 1.3 (Draft), 2007-07-13

2. Using the Editor

Since the introduction of the Windows-Icons-Mouse-Pulldown (WIMP) Graphic User Interface (GUI), tool designers have aimed at achieving an "intuitive" interface, frequently it seems, to avoid having to write documentation like this! This section will cover the operation of the MDE, highlighting areas which may not be obvious at first glance in the hopes that the editor interface actually *is* intuitive.

2.1. General Layout

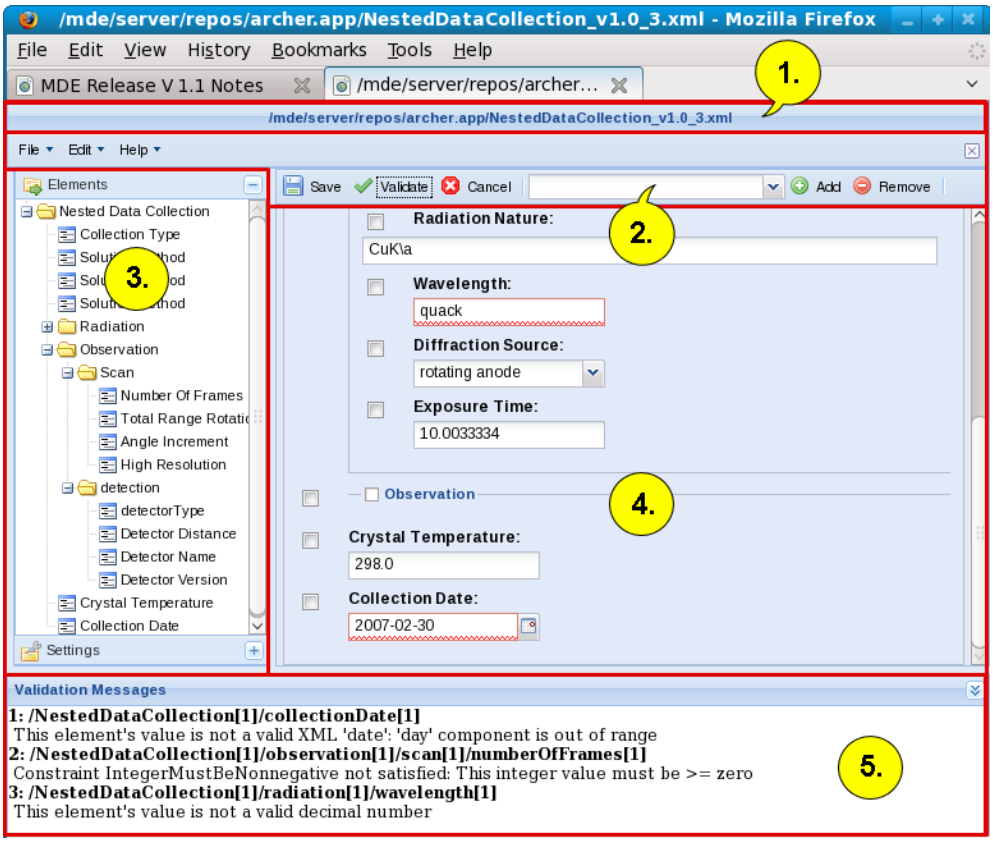

Figure 1 – Conceptual Editor Areas

The editor as displayed in a typical Browser is shown in Figure 1. The details of how some controls are rendered may vary with the browser being used, but the general layout and operation will be consistent. The editor contains five main sections:

1. Title bar provides information about the identifier of the record loaded for editing and its associated schema.

2. The Action selection area comprising a conventional pull-down menu and tool-bar. Most tool-bar actions have corresponding menu items. The menu selection also contain the tool-bat icons as a aid to learning what tool-bar icon performs which function. Note that some actions may only be performed from the menu, while other require the tool bar. For instance, there is no "About..." tool-bar button, and no menu item to insert a new metadata element.

3. The Tree View panel shows the element names of all elements in the record. Positioning the mouse pointer over an element will display the schema description of that element as *hover-text* after a short delay.

4. The actual editable values of the metadata elements depicted by the tree view are shown in the Detail Display panel. When the schema dictates that an element value must be selected from a controlled term list, the field will appear as a combo-box with a drop-down list button. Elements that contain validation errors will be underlined. This is explained in greater detail in Section 2.3.

5. The Validation Message panel is not displayed until the editor detects a condition that breaks a schema defined constraint. This may occur following use of the Validation command, or when saving the record (pre-save validation is implicit in a Save operation).

2.2. Exiting the Editor

In common with most editors, the MDE provides a means of exiting which warns if the record has unsaved changes, and a way of (1) exiting regardless of the record state (2). An Exit can be triggered from either the menu, or the Tool-bar. The appropriate controls are highlighted in Figure 2.

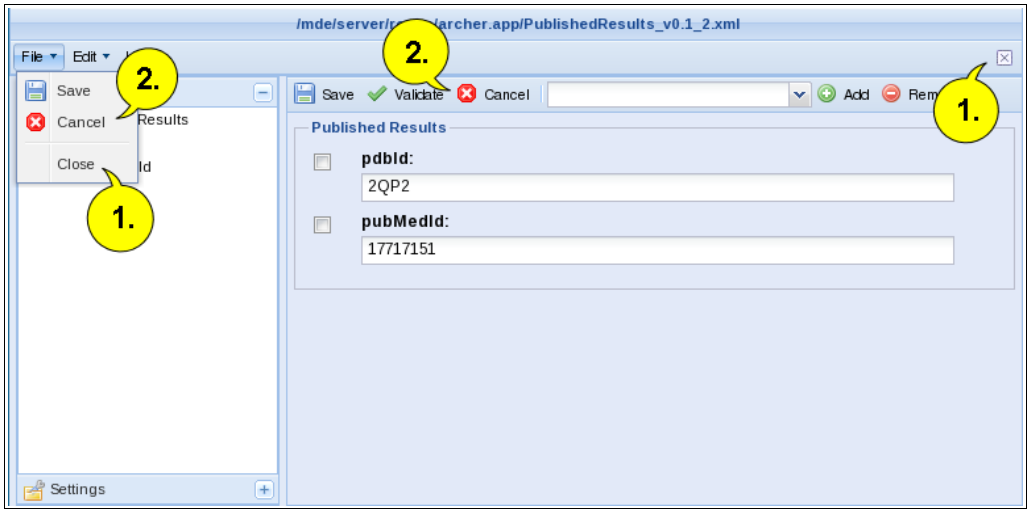

Figure 2 – Close Options

The events associated with an Exit-cancel operation are Application dependant. Some Applications will be able to undo any Save actions issued since the editor was launched on the target records; others will not. Consult the Application documentation in the hope that it will explain how the MDE Edit-cancel operation is handled.

2.3. Validating a Record

Record validation is invoked automatically whenever the record is saved. It may also be preformed manually through the `Edit | Validate` menu item, or the corresponding Tool-bar button. The validation process uses schema defined rules to:

- Ensure all Element names are appropriate for the designated schema
- Check element cardinality (*minoccurs* and *maxoccurs* values)
- Ensure the Element value is of the specified type (when mandated)
- Ensure controlled term values precisely match valid choices
- Apply any cross-validation rules such as element mutual exclusion

If the record passes validation, a pop-up will briefly appear giving this glad news (see Figure 3). Any errors will be reported in a panel that opens at the foot of the browser window. This will list all errors detected and attempt to identify the source and cause of each problem. A vertical scroll bar will appear if there are too many errors to show all within the panel. Where a validation error is due to an element, the offending element will be indicated in the editor panel by a red underline.

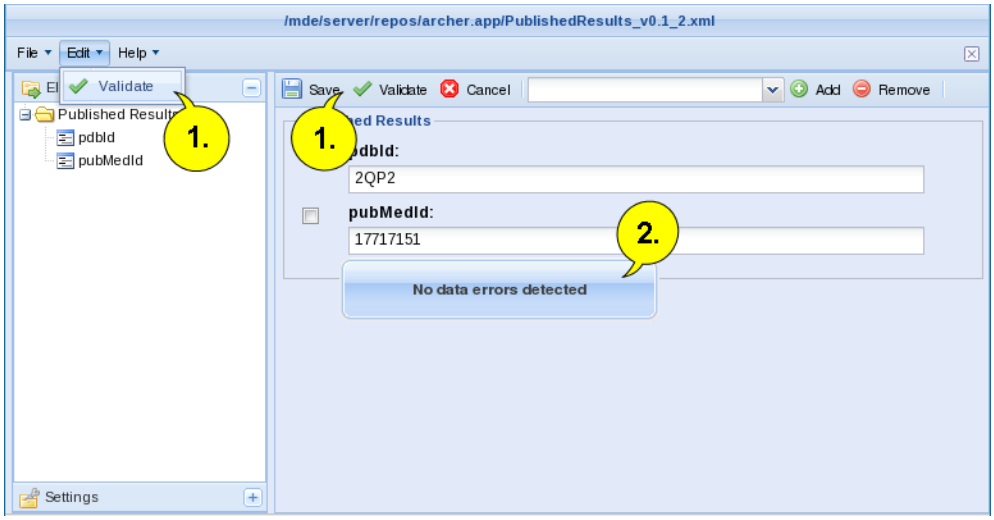

Figure 3 – Happy Days

The red underline which flags an Element with validation problems is cleared as soon as the Element value text field receives the cursor focus. This obviously does not indicate that the error no longer exists. Users should re-validate the record after making changes.

As a General Philosophy, the MDE places the User in charge. This means that the MDE makes it possible for a User to request that a record be saved, even though it does not pass validation. However, the Application that invokes the editor has the last say and may refuse to allow this. Your Application User Guide may, if you are fortunate, state what will happen in this case.

2.4. Saving a Record

Records may be saved at any time by selecting the `File | Save` menu item, or pressing the corresponding Tool-bar button. In common with the generally accepted editor paradigm, saving a record will not close the editor session. To provide the User with visible feedback that their changes to the the record have indeed been made permanent, a brief

popup will appear as shown in Figure 4. As described in the section on Validation, a Save operation is always preceded by a full record validation. If errors are detected, a dialog will appear providing the option to abandon the Save, or to attempt to force it regardless of the record state.

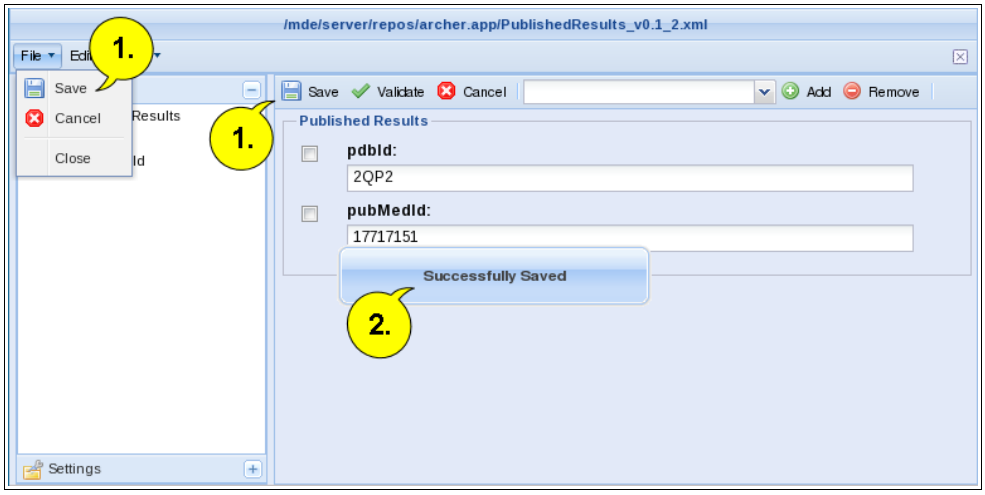

Figure 4 – Saved

As you might expect, trying to save a record with validation errors, or exiting from the editor with unsaved changes will trigger a dialog box that alerts you to the situation and provides the option to abandon the action, or proceed regardless.

2.5. Adding an Element

The point of insertion for new elements is after the current element selection, so select the element before where you want the new one added (1). Elements are added using the combo-box located on the Tool-bar. Open the drop-down list and select the required element to be added (2), then press the `Add` button (3). The new, blank element appears in the tree and detail view panel (4). Although this inserts an element, you must save the record as described in Section 2.4 to make the insertion persistent.

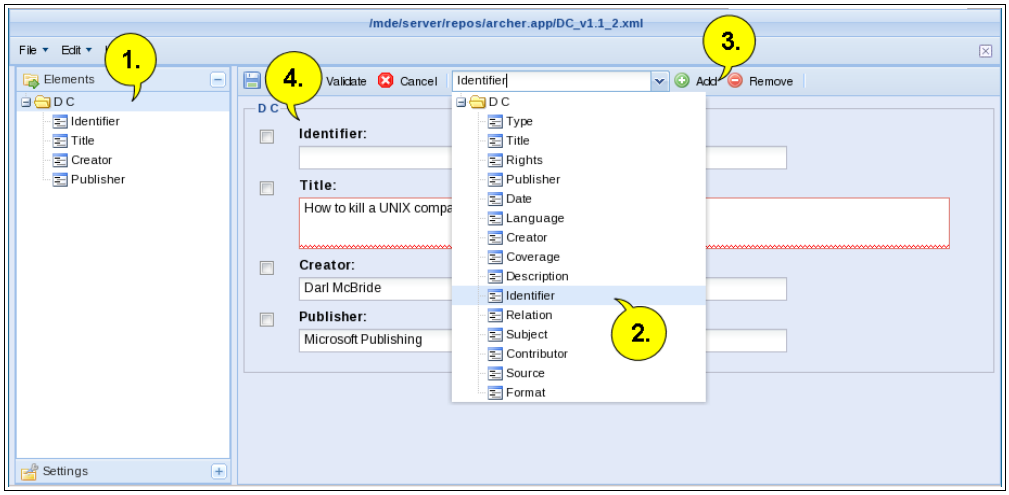

Figure 5 – Add Element

If you highlight an element in the Tree-view panel (1), the new element will be inserted after that element. However as no mechanism exists for persisting the User's preferred element ordering, the element may not appear in this position the next time the record is loaded.

2.6. Nested Elements

Where a schema supports it, one element instance may be contained inside another. This is termed *Element Nesting*. The MDE supports nesting to infinite depth, although practical limitations will apply. Elements may be considered to be of one of two types, *leaf*, or *branch*.

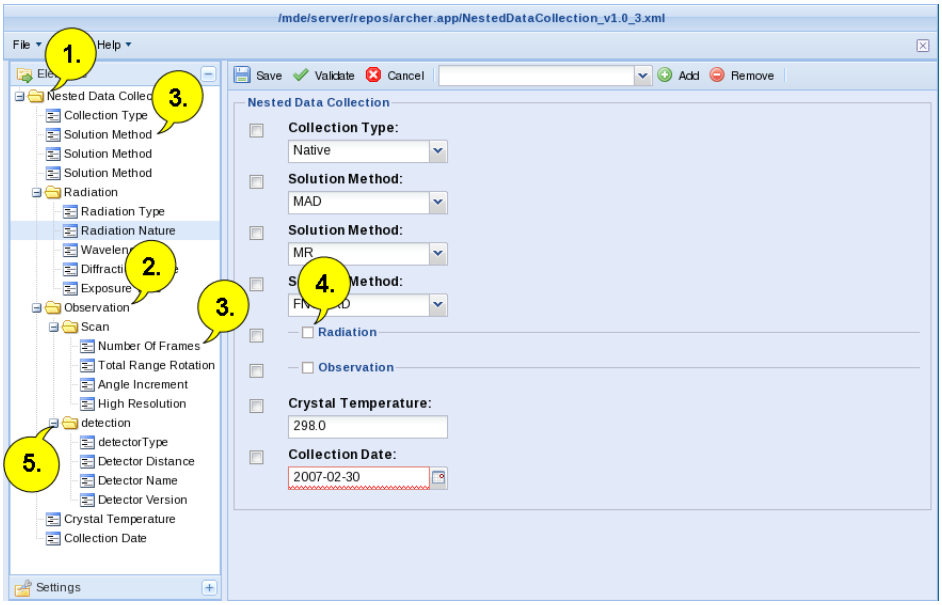

Figure 6 – Nested Elements

A record conforming to a schema containing nested elements is shown in Figure 6. Only *leaf* elements (3) may contain data values. *Branch* elements (2) are depicted in the Tree View panel by a Folder icon. A *branch* may contain any number of either *branch* or *leaf* elements, but not a mixture of both. The Root Element (1) is also depicted in the Tree View using a Folder icon, but it is a special case representing the entire record and so may contain both leaf and branch element types.

On the Detail View panel of the MDE, branch elements have their element name tagged by a check box (4) which enables the contained elements to be hidden. By default, all nested elements are “rolled-up” to hide their contents when the record first loads. This also applies to the Tree View where the roll-up and expand operation is controlled by the Junction icon (5) to the left of the Folder icon. Note that unrolling a branch element in the Tree View panel has no impact on the Display View, although selecting a leaf element of a branch in the Tree View will unroll sufficient of the corresponding structure in the Detail Display panel to make the selected element visible and “current” if it is not already visible. Rolling up a branch in the Tree View does not collapse the Detail Display structure.

2.7. Adding Nested Elements

Figure 7 shows the element selector combo-box for a record whose schema includes nested elements. Nested elements are indicated by a folder icon (1) in both the left hand Tree View Panel and the drop-down element selector combo-box. Leaf elements (2) are denoted by a text panel icon. Note that branch elements may be "rolled up" to hide temporarily hide their contents using the associated check-box (3) in the detail view, or the tree node junction icon (4) in the Tree View panel and Element Selector combo-box.

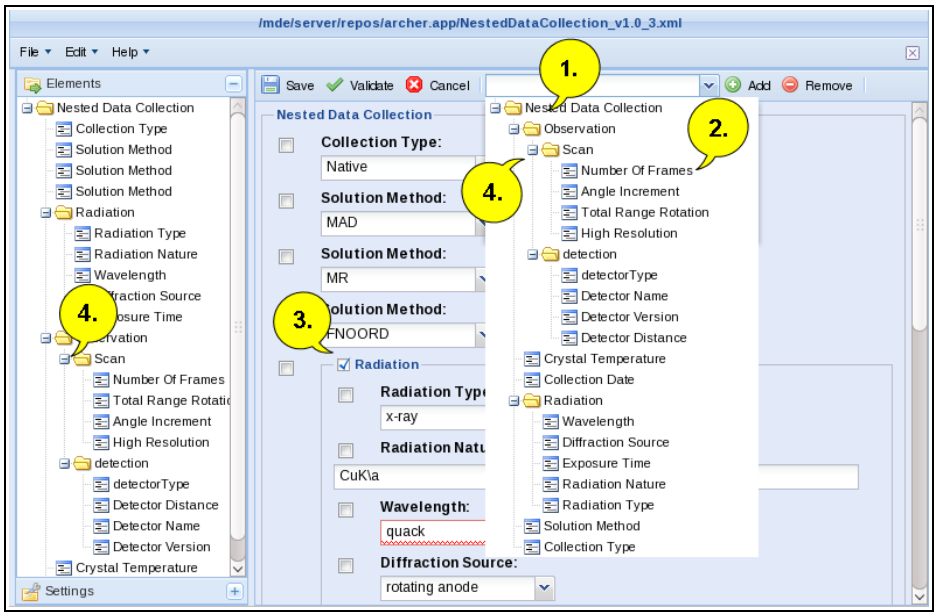

Figure 7 – Nested Elements

All metadata element instances with an associated value are *leaf* elements. They cannot contain other leaf or *branch* elements. A *branch* element may contain any number of other *branch* or *leaf* element instances, but not both.

Metadata schemas which include nested elements define which elements may appear inside others. The overall schema structure is represented in the Element Selector combo-box. When elements are added to a record, the MDE uses this structure and the "current" selected element of the record to determine where to create the new element instance. It is not possible to insert an element in a place that will violate the structure defined by the schema, however there may be cases where the MDE is unable to decide where to place a new element because multiple possibilities exist. The rules for where new instances will appear are similar to those for leaf elements:

1. If an element is highlighted in the Tree View, a new *branch* node will be created below the first element where the branch element may legally appear. That is, the MDE will not permit a branch element to be created inside another branch element unless the schema

permits it. Attempting to do so will cause an error message pop-up to appear.

2. If no element is highlighted, new branch elements will be added at the top of the tree.

3. If the element being inserted is a *nested leaf* or *branch* and either no element of the Tree View is selected, or a *leaf* element directly under the root element is highlighted, the MDE will create the required nest of element instances required to contain the new element.

4. If the element being inserted is a *nested leaf* or *branch* and the current, selected element is a valid containing element for this element type, it will be inserted inside the highlighted element at the appropriate place, creating any containing *branch* elements if required.

5. If the MDE encounters a situation where it has more than one choice available of where to place a nested *leaf* or *branch* element, it will display an error pop-up dialog requesting the user to be more explicit and highlight (select) an existing element location such that there is only one choice available to the MDE for where the new instance should be placed.

In summary, MDE will attempt to insert new leaf and branch elements in accordance with the structure mandated by the schema so long as the current location as denoted by the element selection highlight is unambiguous. If the selection results in ambiguity, or would violate the record structure defined by the schema, the insertion will not be allowed and the user notified by a pop-up message.

2.8. Deleting an Element

Select the Element or Elements to be removed by checking the *checkbox* located to the left of the Element in the detail view panel (1). This is shown in Figure 8. After selecting the element, or elements to be deleted, press the `Remove` button (2) to delete them from the record loaded in the editor. Naturally, the deletion will not be persisted until the record is saved as described in Section 2.4.

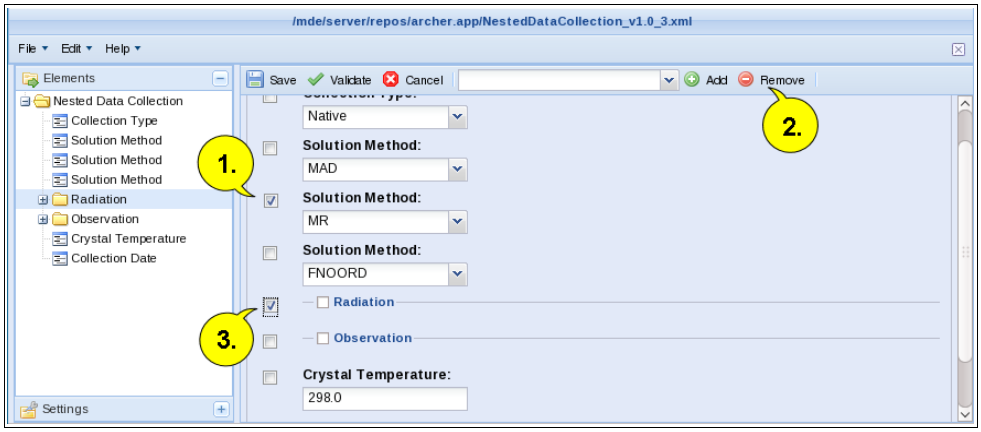

Figure 8 – Element Removal

Deleting nested elements is performed in the same way for *leaf* elements. If the target element is a *branch* element (3), deletion obviously must also remove all elements contained within the branch as well.

Adding or removing an Element may cause the record to become invalid if the new state breaks cardinality rules enforced by the schema. As noted earlier, you may re-validate the record at any time, and in any case, the record will always be revalidated before it is persisted by a `Save` action.

Deleting a *branch* Element will delete all elements contained within that branch. This cannot be undone and no warning is given.

2.9. Working with Templates

Experience has shown that Users find the process of creating a new metadata record from just a schema to be painful. This distress can be mitigated if they are able to use a Template. This is an essentially invalid record comprising the elements commonly used in records of the type, perhaps with some values pre-set, but the others blank. The MDE supports this feature provided some provision for it has been incorporated into the Application which presents the User with a Template selection. The Application may even support User modification of existing Templates.

In operation, Template editing is indistinguishable from normal record editing. The only difference comes about when the record is saved, at which time the changes are written to a new record identifier thus ensuring that the Template remains unaltered and ready for re-use.

2.10. Help Me!

The `Help | Release Notes...` menu item provides access to the Release Notes applicable to the version of the MDE in use (see Figure 10). `Help | About...` opens a dialog box which lists information that is important when reporting problems to the development team (see Figure 9).

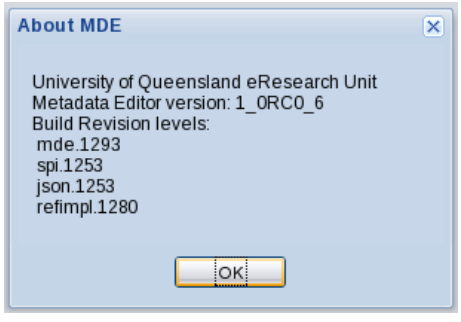

Figure 9 – Help About

The `Help` menu also contains an option to load this User Guide from the server running your Application. This action currently takes the form of a Portable Data Format (pdf) document download, so depending on your web

browser settings, you may need to perform additional actions to open the manual for viewing.

3. Capabilities and Restrictions

This section provides an overview of the basic capabilities and restrictions of the MDE. For more precise details including details of known defects and work-around solutions, please consult the Release Notes that accompany the actual editor. These are available on-line through the `Help | Release Notes...` menu item (see Figure 10).

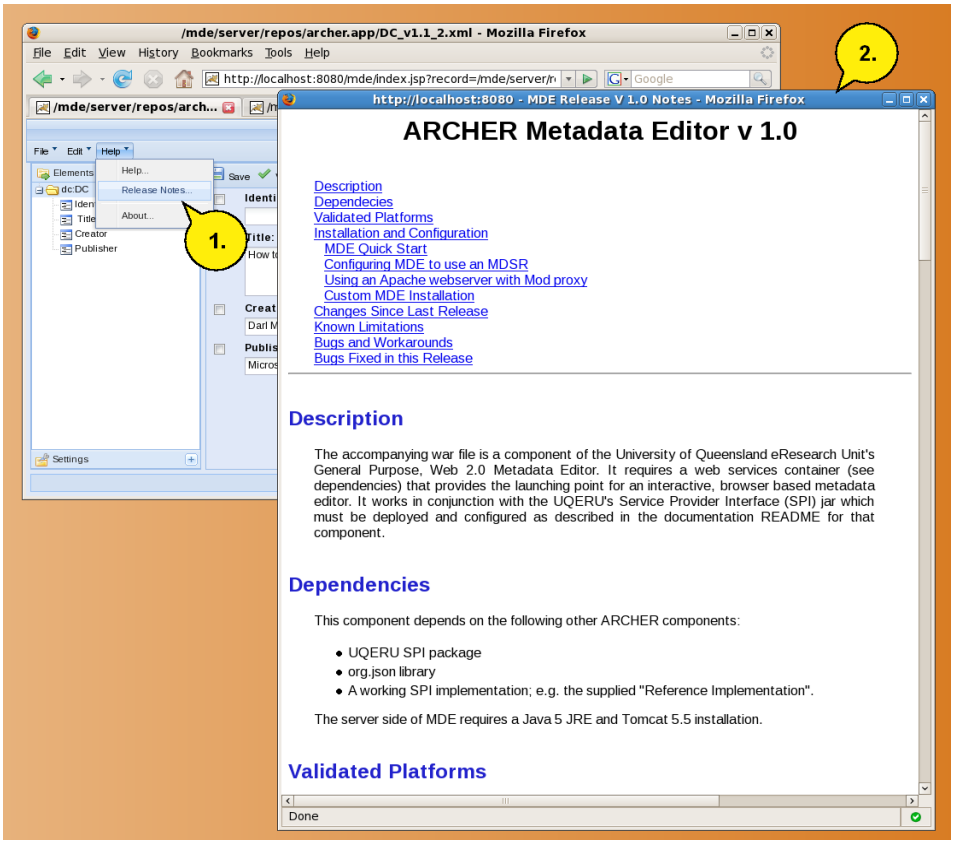

Figure 10 – Viewing the Release Notes

3.1. Element Ordering

The order in which elements appear in the editor is that in which they appear in the underlying schema. As there is no universal meta-metadata that would allow a "user-preferred" order to be persisted, changing the order is not supported.

3.2. Printing

The editor has no print capability.