Dynamic Linked Cells:
An Implementation of "Virtual Objects" in DOORS

Michael Sutherland
Galactic Solutions Group, LLC
Michael.Sutherland@galactic-solutions.com

Prepared for the Telelogic Americas 2006 User Group Conference

23 October 2006
Abstract:
DOORS has the ability to link an object in one module to an object in another module. The text and attributes from the other object can be viewed in a separate column via "Layout DXL", but cannot be incorporated into the main column of the DOORS module. A technique called "Dynamic Linked Cells" will be demonstrated which allows the addition of "Virtual Objects" into the main column of a DOORS module. This technique has the following benefits:

(1) Cell is automatically updated when the data is changed at the source.
(2) Cell appears in "main" column, and data exports to Word or Excel.
(3) Cell is not editable, and the data can only be changed at the source.
(4) Data is mastered in one location and can be referenced in multiple locations.
(5) Cells do not update after baselining, preserving the state of the data.
(6) Cells retain their data when the module is archived and moved to another database.
(7) Multiple attributes can be shown in the same cell.

The technical details of the development of the "Dynamic Linked Cells" application will be discussed, and the full DXL source code for the implementation will be made available to attendees.

Author Biography:
Michael Sutherland is the founder of Galactic Solutions Group, and has 16 years experience working with the automotive and military industries. Michael was a consultant to General Motors Powertrain and Manufacturing for 11 years. He is currently working with General Dynamics on the Future Combat Systems project, developing and deploying Systems Engineering processes and tools. Michael has a Masters Degree in Electrical and Computer Engineering from Oakland University in Rochester MI. He also specializes in the application of the DOORS database, mentoring and teaching application, customization (DXL), and information modeling to a wide variety of clients across the United States.
Background:

On a complex product development project, high-level requirements for the system are developed and decomposed into derived requirements for subsystems and components. Lower-level specifications for requirements, design, and interface specifications need to reference data from high level specifications.

Figure 1: Product Specifications

DOORS uses links to associate data in one Module with data in another Module. Layout DXL in a column can “pull” the data from the one module to the other dynamically. This data does not appear in the DOORS “main” column. Instead, the data appears in a separate column. The data exports to Word “Book” form as an attribute, in the following format:

“Layout Column Name : Layout Column Data”

Assigned “Paragraph Style” is ignored for Word style formatting.

Since this data is not actually written to the module, DOORS archives do not contain the data when they are restored in another database.
“Dynamic Linked Cells”

To solve these issues, a technique called “Dynamic Linked Cells” was developed in DOORS to allow “Virtual Objects” to be created that dynamically update their content by following an out-link to a target object, and copying the data to a DOORS table cell in the DOORS “main” column.

Application: Managing SWaP-C Data

SWaP-C (Size, Weight, Power, and Cooling) allocation and estimate data is tracked for each subsystem and component of the product, and is referenced in many specifications. Placing this data in single DOORS module, and using Dynamic Linked Cells to reference the data, allows each specification to be automatically kept up to date and synchronized with the master SWaP-C data table. The SWaP-C data table is composed of rows and columns, where the rows are the subsystems and components which are represented as DOORS objects, and the columns are the SWaP-C data which is represented as DOORS attributes.

![Figure 2: SWaP-C Weight Allocation Table](image)

Implementation:

The goal is to implement a method to allow specifications to reference data “cells” in DOORS modules, and keep the data updated.

The implementation of “Dynamic Linked Cells” combines three unique features of DOORS, Attribute DXL, the Attribute property of DOORS Tables, and Link Attributes.
Attribute DXL:

DOORS allows “Attribute DXL” code to be associated with an attribute. Attribute DXL updates the values of the attribute for objects in the module when a module is opened, or when a user selects “Tools -> Refresh DXL Attributes” in a Module window. Attribute DXL uses two pre-declared variables (attrDXLName, obj) to determine what attribute is being updated, and what object is being updated for each execution of the code:

```
attrDXLName
```

Declaration

```
object obj

const string attrDXLName
```

Operation

DOORS attribute programs run in a context where the variable `obj` is already declared to refer to the object whose attribute is being calculated.

The constant `attrDXLName` can be used instead of a literal attribute name to refer to the attribute value that is being calculated. This allows one piece of DXL attribute to be used for several attributes without being modified.

Example

```
obj.attrDXLName = today
```

Table Properties - Attribute:

A DOORS table is composed of DOORS objects that have attributes. A DOORS table has a property that specifies the value of the attribute to be displayed in the table cell. This attribute can have associated Attribute DXL.

![Figure 4: DOORS Table Properties – Attribute Tab](image)

Attribute DXL can be written to follow a link out of a table cell, and copy data from the target Object. This technique was used to create an implementation of “Dynamic Linked Cells.” A DXL attribute called “Cell Text” is created.
Figure 5: Attribute Definition for Cell Text Attribute

Figure 6: Attribute DXL inclusion for “Cell Text” Attribute
Each “Dynamic Linked Cell” uses this “Cell Text” attribute.

Figure 7: Table Properties and “Cell Text” Attribute

To facilitate ease of use when creating a “Dynamic Linked Cell”, “Insert Dynamic Linked Cell” DXL functionality has been created to automate these steps for the user. This DXL functionality creates the cell, removes cell borders, sizes the cell, and sets the attribute property for the table to the “Cell Text” attribute.

Figure 8: Insert Dynamic Linked Cell – Insert Menu Functions
Figure 9: “Dynamic Linked Cell” – Before Insert

The automatic removal of the cell borders makes the cell appear native to the DOORS “main” column.
Once the “Dynamic Linked Cell” is inserted, the user creates a link to the target object. Refreshing DXL attributes will perform the initial pull and population of the data.
Figure 11: “Dynamic Linked Cell” after link creation

Figure 12: SWaP-C – Referenced Row after link creation
**Attribute Selection:**

A user of “Dynamic Linked Cells” may want to reference more data than just the DOORS “Main” Column (Object Heading and Object Text). To achieve this, Link Attributes are used to store the attribute names.

Link Attributes can be created from the “Edit -> Attributes…” menu in a Link Module:

![Figure 13: Link Module – Create Link Attributes](image)

A link attribute named “Target Attributes” is defined and used in link modules containing links from “Dynamic Linked Cells” to the referenced target data. The link references the entire row, and the attribute name allows the “Dynamic Linked Cell” to target a specific “Cell” (object/attribute combination) in a DOORS module.

![Figure 14: Link Module – Edit Link Attribute Value](image)
Editing link attributes through the link module is a complicated, multi-step process.

Steps:
1. Locate and open link module
2. Locate link
3. Right-click on link; select "Edit Link"
4. Select Attribute, "Edit Attribute Value" box will appear
5. Set "Value " toggle to "Specific"
6. Type attribute name(s) in text field.
7. Click "Apply" on "Edit Attribute Value" box
8. Click "Close" on "Edit Link Object" box
9. Close Link Module

Figure 15: Edit Link Attribute Value – Link Module

Editing link attributes through the source module is a complicated, multi-step process.

Steps:
1. Select cell
2. Select "Link -> Edit Links..." from menu
3. Select Link; click "Details..."
4. Select Link
5. Click "Details..."
6. Select Attribute
7. Click "Edit..."
8. Type attribute name(s) in text field
9. Click "OK" on "Attribute Value" box
10. Click "OK" on "Link Properties" box
11. Click "OK" on "Object Properties" box

Figure 16: Edit Link Attribute Value – Source Module
Set Attributes… Function:

Because editing link attributes from the DOORS user interface is a complicated, multi-step process, and typing attributes names is prone to error and inconsistency, a “Set Attributes…” function was developed as part of the Dynamic Linked Cell implementation.

Figure 17: Menu Selection – “Dynamic Linked Cell -> Set Attributes…”

The “Set Dynamic Linked Cell Attributes” function scans the target module, and displays a list of attributes available in the target module.

The user can then select attributes and “push” them over to the “Displayed Attributes” list. This list determines the attribute content and order for the “Dynamic Linked Cell”.

Figure 18: “Dynamic Linked Cell -> Set Attributes…” – Attribute Selection
Figure 19: “Dynamic Linked Cell -> Set Attributes…” – Attribute Selected

Figure 20: “Dynamic Linked Cell -> Set Attributes…” – Attribute Displayed
Once the dialog box is closed, the updates will be applied, and the new data will be pulled. Attributes names will appear in bold, followed by the attribute value.

![Dynamic Linked Cell - Attribute Display](image)

Figure 21: “Dynamic Linked Cell” — Attribute Display
A special attribute called “Object Heading and Object Text” is available in the list of available attributes. This is to pull the data from the DOORS “main” column, and is represented without an attribute label. This can be commingled with other attributes in any order.

Figure 22: “Dynamic Linked Cell -> Set Attributes…” – “Object Heading and Object Text” Selected

The “Object Heading and Object Text” attribute can be “pushed” over to the Displayed Attribute list using the left arrow.

Figure 23: “Set Dynamic Linked Cell Attributes” – “Object Heading and Object Text” Displayed
When more than one attribute is displayed, individual attributes can be selected and reordered to in the list using the up and down arrows.

![Figure 24: “Set Attributes..” – “Object Heading and Object Text” Displayed First](image)

“Set Attributes..” – “Object Heading and Object Text” Displayed First
For SWaP-C data, the “Object Heading and Object Text” contains the subsystem or component name. Pairing that with the attribute containing the data for a specific variant gives a complete representation of the data.

Figure 25: “Dynamic Linked Cell” – “Object Heading and Object Text” and one additional attribute displayed
SWaP-C data has units (Weight == kg, Power == Watt, etc.). The SWaP-C module stores these units as Module level attributes, one for each data column. The “Dynamic Linked Cells” use this knowledge to append the units to the data when it is pulled into the module.

Figure 26: SWaP-C RAM-T Module level “Units” Attributes
Automatic Updates with Attribute DXL:

Since “Dynamic Linked Cells” are implemented with Attribute DXL, they automatically update when a module is opened, or when a user selects “Tools -> Refresh DXL Attributes” in a Module window.

In DOORS v7.x, Attribute DXL refreshes data even if the Module is opened in read-only mode, allowing read-only users to see the latest data. This is desirable when the read-only module being opened is the “current version” of the Module, but is not desirable when the read-only module is a baseline version of a module.

Because of this feature, the “Dynamic Linked Cell” functionality also copies the information contained in the “Cell Text” attribute to the standard “Object Text” attribute. This facilitates the following operations:

- **Baselines:**
  
  Data in a DOORS baseline should not change, but Attribute DXL is executed in baselined modules, and Attribute DXL cannot be disabled or turned off once the module is baselined.

  To circumvent this problem, the “Dynamic Linked Cells” Attribute DXL code was written to detect if it is being executed in a baseline. In such a case, the Attribute DXL does not update the attribute value. Instead, it displays the value stored in “Object Text”.

- **Archives:**
  
  DOORS Archives are used to move entire DOORS modules between databases. Module Archives are for a single module, and do not carry any link information with them. To allow data pulled into a “Dynamic Linked Cell” to be persistent, a DXL script is used to disable the “Cell Text” attribute for “Dynamic Linked Cells”, and the tables are reset to show “Object Text”.


**DOORS Table Property: “Value displayed in each table cell”**

The DXL used to set the Table Property of the “Value displayed in each table cell” is the undocumented function (reserved “Main Column Attribute”).

To set the value displayed in the cells of a table to the “Cell Text”, attribute, the following DXL is executed on the Table Header Object:

```dxl
if ( table( o | ) ) {
    o.(reserved "Main Column Attribute") = "Cell Text"
}
```

**Figure 27: Using DXL to enable “Dynamic Linked Cells”**

To set the value to the default of “Object Heading and Object Text”, a null string (""") value is assigned for all Table Objects.

```dxl
Object o
for o in entire current Module do {
    if ( table( o ) ) {
        o.(reserved "Main Column Attribute") = ""
    }
}
```

**Figure 28: Using DXL to disable “Dynamic Linked Cells”**
DOORS Tables

In the initial implementation of “Dynamic Linked Cells”, the cells were intended to stand alone as a single cell. Since the adoption of “Dynamic Linked Cells”, users have expressed a need to build a larger table comprised of “Dynamic Linked Cells”. This update is being evaluated, and may be implemented if feasible.

<table>
<thead>
<tr>
<th>Component</th>
<th>Space</th>
<th>Weight</th>
<th>Power</th>
<th>Cooling</th>
</tr>
</thead>
</table>

Figure 29: Multiple “Dynamic Linked Cells” in a single DOORS Table

“Virtual Objects” in DOORS

The concept of creating self-updating, “virtual objects” in DOORS has been previously conceived and implemented by others (see Appendix A). The main advantages of the “Dynamic Linked Cell” concept are:

- Since Attribute DXL is used, no Triggers are necessary.
- Since Attribute DXL in v7.x shows updated values in read-only mode, users see the latest data in all module open modes.
- The “Set Attribute…” tool allows a user to easily select from the available set of target attributes.
- Referenced attributes appear in the DOORS “main” column, and are exported with the “main” column to Excel or Word “Book” form using the standard DOORS exporters.
Appendix A: Other “Virtual Object” implementations in DOORS:

- Dynamic and Automatic Update of Content Linked from Another Module Using DXL Triggers
  - by John Deely and Nick Nicolai of GE Healthcare Technologies
  - Prepared for the 2005 Telelogic Americas User Group Conference

- Comparative Evaluation of Product Line Requirements Architectures
  - Developed at Honeywell by Robert H. Fall, Jennifer Stieglitz, Robert Schlachter, Karen Protola
  - Presented at the 2001 Telelogic Americas User Group Conference

- Virtual Objects? Virtually!
  - by E. M. O’Keefe – Raytheon Company
  - Presented at the 2000 INDOORS Americas User Group Conference

Appendix B: “Dynamic Linked Cells” Download and Installation:

The “Dynamic Linked Cells” software can be downloaded from the following URL:


The downloaded .zip file will contain the following files:

- Dynamic Linked Cell\cellText.inc (Attribute DXL)
- Dynamic Linked Cell\Insert\Dynamic_Linked_Cell_Insert_After.dxl
- Dynamic Linked Cell\Insert\Dynamic_Linked_Cell_Insert_Below.dxl
- Dynamic Linked Cell\Insert\Insert.hlp
- Dynamic Linked Cell\Insert\Insert.idx
- Dynamic Linked Cell\Dynamic Linked Cell.hlp
- Dynamic Linked Cell\Dynamic Linked Cell.idx
- Dynamic Linked Cell\Dynamic_Linked_Cell_Set_Attributes.dxl

Extract these files to the “user” directory within the DOORS client installation. The following example location is for a DOORS 7.1 installation:

C:\Program Files\Telelogic\DOORS 7.1\lib\dxl\addins\user