INTERGRAPH

SOFTWARE SOLUTIONS

A PROPOSED IMPLEMENTATION OF THE RASSP CONFIGURATION MANAGEMENT MODEL USING INTERGRAPH DM2

INTERGRAPH
Intergraph Corporation
Huntsville, Alabama 35894-0001

Date: May 31, 1996

TABLE OF CONTENTS

ACRONYMS AND ABBREVIATIONS

CM	Configuration management
DM	Document Management
EF	Enterprise Framework
EFCM	Enterprise Framework Configuration Management
GUI	Graphical user interface
RASSP	Rapid Prototyping of Application-Specific Signal Processors
SQ	Saved query
TO	Transfer ownership
VL	Vault location
WL	Work location
WS	Workspace

LIST OF FIGURES

1. Workspaces

1.1 Shared and Private Workspaces

Workspaces are partitions of the design object space to allow designers working on the various parts of a project to selectively make their design objects visible to others in the project [Cattell, 1991]. In the RASSP Configuration Management (CM) model, three types of workspaces exist: private, shared, and global. Workspaces are organized in a hierarchical fashion as shown in Figure 1.1. Each node in the hierarchy represents a workspace. Branches in the hierarchy represent a parent-child relationship between workspaces. The *global workspace* is at the root of the hierarchy, *shared workspaces* are the intermediate nodes in the hierarchy, and *private workspaces* are the leaves in the hierarchy. [Martin Marietta, 1994]

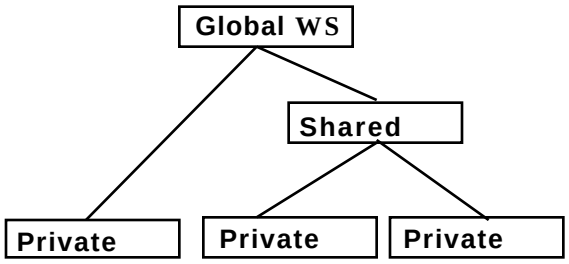

Figure 1.1 Workspace Hierarchy

1.2 Workspaces in DM2

The above workspace hierarchy can be implemented in DM2 using the features of *users* and *vaults*. Relationships between workspaces may be enforced by defining *groups*, which contain related users, and limiting the access of these groups through the use of *rules*.

In DM2, each user has a private workspace. That is, there is a one-to-one mapping between a user and a private workspace. Rules may be used to enforce the privacy of individual workspaces. Shared workspaces can be implemented through the use of vaults. A vault is a logical collection of shared objects. Rules can be used to control access to a vault. User-to-vault relationships can be established to allow visibility of ancestor workspaces. The global workspace will consist of selected data from all shared workspaces (vaults), obtained through the DM2 *query* capability.

The proposed DM2 implementation of the RASSP Enterprise Framework workspace hierarchy is shown in Figure 1.2.

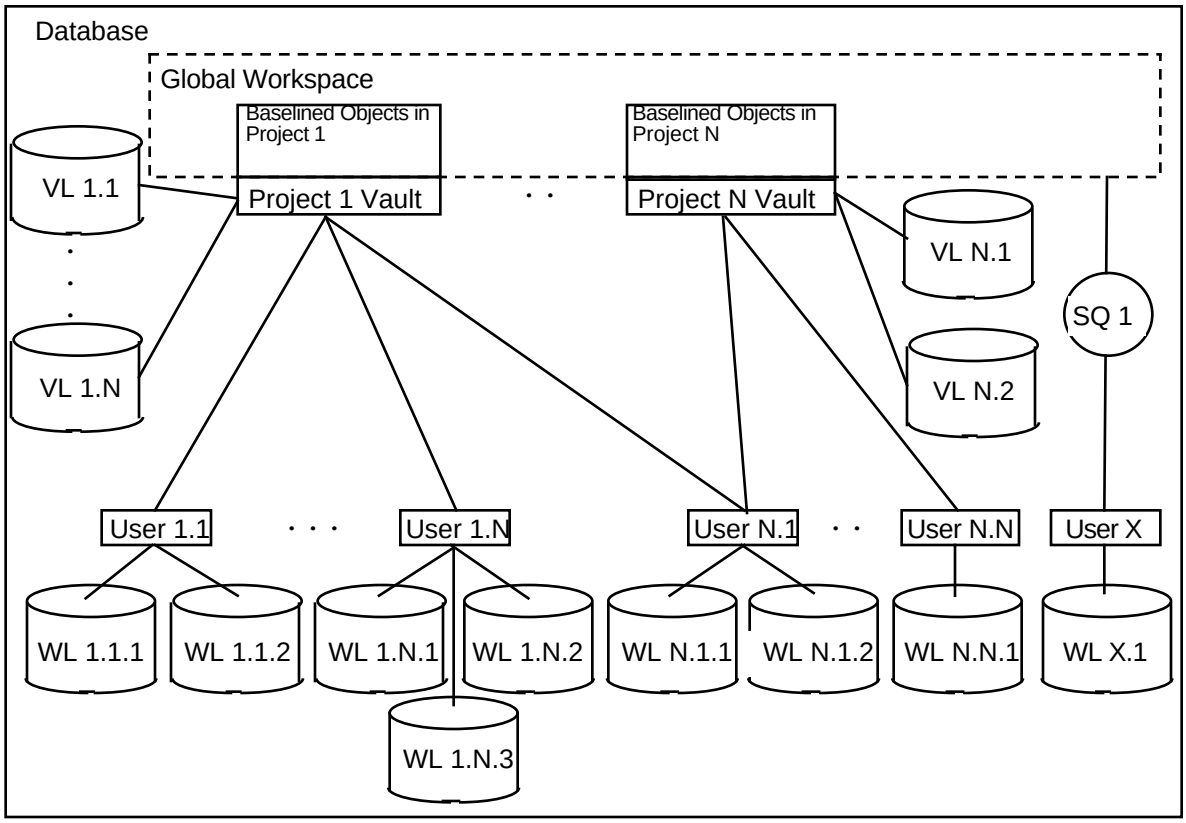

Figure 1.2 Enterprise Framework Workspace Hierarchy *****

***** Legend:

= Represents the physical file system location for the items residing in the associated vault (shared workspace) or user (private workspace). Vault and work locations may be on the same host machine or on separate machines within a network.
VL = Vault Location: Each vault location will be associated with only 1 vault.
WL = Work Location: Each work location will be associated with only 1 user.

= Represents a DM2 saved query.
SQ = Saved Query: Every user will have access to the Global Workspace through a predefined query.

= Represents a logical collection of objects based on ownership. An owner of an object is either a user or a vault.
Each vault may have 1 or more vault locations associated with it.
Each user may be associated with 1 or more vaults.
Each user will have 1 or more work locations associated with it.

*The version numbers used in the above figure do not reflect the DM2 naming convention, but are used to provide clarity to the figure***.**

2. Data Object Management

2.1 Configurations and Version Control

The RASSP CM Model uses a data object versioning scheme where related data objects that evolve at the same time are grouped together as a *configuration*. At any point in its life cycle, a configuration can exists in one of three states: *transient*, *working*, or *released*. Upon creation, a configuration is considered to be transient and is associated with a private workspace. A transient version of a configuration allows for updates and may be deleted. A transient version may be promoted to a working version when the configuration has reached a level of maturity such that it can be shared with other users. Working versions reside in shared workspaces. At this state the configuration cannot be updated, but may be deleted. A configuration is considered to be in the released state when a working version of that configuration is promoted to the global workspace. Released configurations cannot be updated or deleted. [Martin Marietta, 1994]

A transient version of a configuration may be created from a previous version regardless of its state. The source configuration remains unchanged if it is a working or released version. Creation of a transient version from an already existing transient configuration causes the source configuration to be promoted to the working version level. A configuration may be deleted if it is at the transient or working version level and is at the lowest level in a workspace hierarchy. [Martin Marietta, 1994]

2.2 DM2 Strategy For Version Control

The RASSP Enterprise Framework scheme for version control can be implemented using DM2 features. DM2 is capable of managing individual objects as well as collections of objects. Hence, a configuration, as described in Section 2.1, may be a group of one to many objects. For the purpose documenting an implementation of the Enterprise Framework Configuration Management (EFCM) Model in DM2 a *folder* will be used as an example of a configuration. A folder is a mechanism for grouping a set of objects together for a specific purpose. The objects may be of different classes and may be added or removed as required.

A folder will be considered transient if it is created by a user. At this point, the folder may be updated or deleted. A transient folder may be created in one of the following ways--

- initially, by the folder option in the DM2 *Object --> New* menu in a user's workspace
- a *copy* action on an existing folder, resulting in a newly named folder, regardless of the state that the source folder is in
- a *check out* action on an existing working folder in a vault

· a *revise* on a released folder in a vault.

Transferring ownership of a folder from an individual user to a vault "promotes" the folder to a working version. A working version of a folder may be checked out, baselined, or deleted. A released version of a folder will result from *baselining* a working version. A released folder may not be updated or deleted. *Revising* a folder generates a new copy of the folder which can then be manipulated. Therefore, the global workspace will consist of all baselined Enterprise Framework Business Items, and may be accessed by a predefined query. All users will have the ability to exercise this query.

Figure 2.1 graphically depicts the DM2 version scheme.

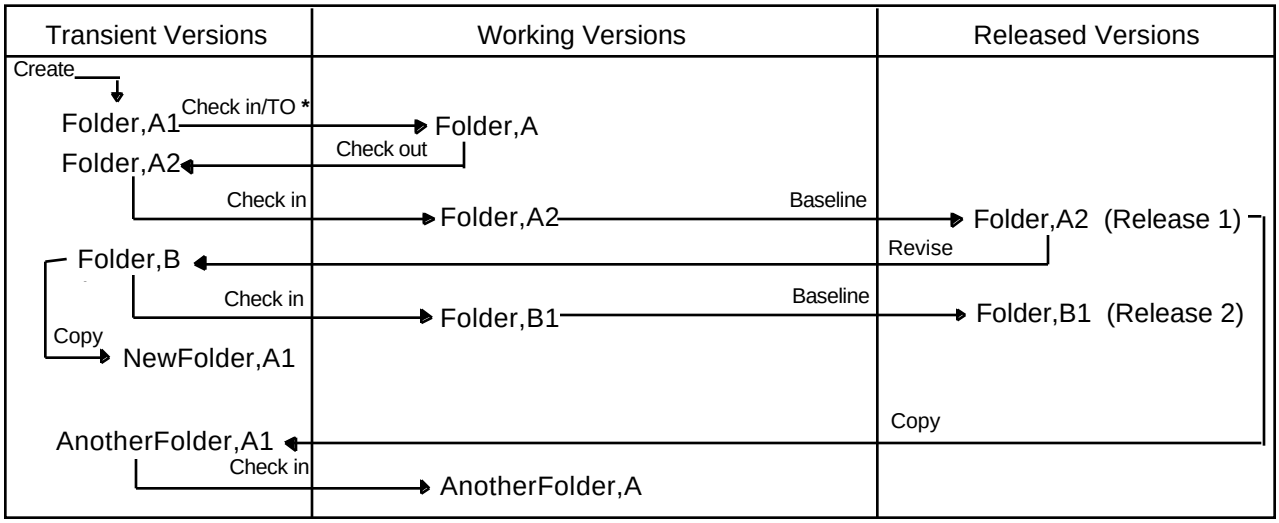

Figure 2.1 DM2 Implementation of Data Item States

***** Transfer Ownership (TO) may be used to promote a folder to a working version the first time.
****** DM2 uses a letter-number combination to reference a folder's Revision and Sequence respectively.

3. CM Functions

3.1 Workspace Functions

A global workspace in DM2 is mapped to all baselined Enterprise Framework Business Items in all vaults within a database. Items are visible in a global workspace by use of the DM2 Saved Query Object Class.

A *shared workspace* in DM2 is mapped to a *vault/vault location* and can be accessed by performing transfer, check in, and check out operations. A vault is a logical collection of shared objects. A vault may contain data objects or actual file system items. A vault location provides a file system location for storing physical files owned by the vault.

In DM2, each user has a *private workspace.* Based on projects, a parent-child relationship can be established between private and shared workspaces. A

private workspace may have more than one *work location.* Similar to vault locations, a work location provides actual file system space for objects residing in a private workspace. Note: A private workspace may have relationships with more than one shared workspace.

3.1.1 Creating a workspace

In DM2, a workspace is created by default upon creation of a user. Shared workspaces are created upon creation of a vault/vault location. Rules dictate parent-child relationship between private and shared workspaces. The global workspace is composed of all baselined objects within all vaults contained in the database. The following steps/concepts serve as a guide in developing the DM2 implementation of workspaces within the EFCM model--

- administration authority is required to create users and vaults (i.e., private workspaces and shared workspaces)
- the global workspace is dynamic in nature and evolves as objects are baselined in project vaults
- vaults will be created with the *noreplace* attribute set to prevent the overwriting of shared data items
- work locations must be created by workspace owners due to write privileges enforced by the operating system.

3.1.2 Accessing an arbitrary workspace

In DM2, the ability to access an arbitrary workspace is controlled by the underlying rules in relation to the requester. Accessing an arbitrary workspace could involve no more than performing a query on that workspace to see what items exist in that workspace. It may also involve copying, updating, transferring, check in/out, baselining, or revising items. Each action is controlled by rules in relation to project, vault, role, and group assignments. The following steps/concepts will guide the DM2 implementation of accessing workspaces within the EFCM model--

- super users will have access to all workspaces within the system
- access to shared workspaces (vault/vault locations) will be restricted based on need (i.e., need to query an item, need to update an item, need to baseline, or revise an item)
- access to private workspaces will be restricted by the rules that control the actions available to non-owners
- all users will have access to the global workspace (defined within the context of a database).

3.1.3 Accessing child workspaces

In DM2, the workspace hierarchy is implemented as relationships defined between an individual user's private workspace and that user's ability to access shared workspaces (vault/vault locations). Shared workspaces will be allowed as parents of private workspaces, and can have more than one child workspace (that is, more than one user) at a time. The ability to access child workspaces will be provided through the DM2 query method, which will return a list of users who have access to a shared workspace. This data is for information only since a private workspace cannot be a parent and a shared workspace will not have more than query access to a private workspace.

3.1.4 Accessing the parent workspace

The DM2 implementation for accessing a parent workspace is much the same as accessing an arbitrary workspace. Rules will limit the scope of access to ancestor workspaces residing between the current workspace and the global workspace. The following DM2 mechanisms will be available to use when accessing parent workspaces: query, copy, update, transfer, check in/out, baseline, and revise. Actions will be controlled in relation to project, vault, role, and group assignments. The following steps/concepts will guide the DM2 implementation of accessing a parent workspace within the EFCM model--

- super users will have access to all workspaces within the system
- access will be restricted based on the relationship between the current workspace and the desired parent workspace
- all users will have access to the global workspace (defined within the context of a database).

3.1.5 Making a workspace visible

The EFCM model calls for the ability to link an application to a specified workspace. The application would then have access to all items in that workspace and its ancestor workspaces. This can be accomplished through tool encapsulation within DM2. Once a tool (application) has been encapsulated, it may be launched via the graphical user interface (GUI) double-click or drag-and-drop capabilities. Rules will govern which users have access to certain applications. Once the application is active within the context of a workspace, the application may manipulate all associated data items in the current workspace and its ancestor workspaces.

For example: An AutoCAD application would be able to manipulate Drawing files residing in work and vault locations associated with the current private workspace and its associated shared workspaces.

3.2 Version Management Functions

The Enterprise Framework concept of a configuration will be implemented in DM2 as a folder.

3.2.1 Creating a configuration

In DM2, creation of a folder is achieved through the point-and-click capability of the GUI. When created, the state of a folder is transient (i.e., it will reside in a user's private workspace). The following steps/concepts will guide the DM2 implementation of creating a configuration within the EFCM model--

- super users will be allowed to create folders in any available private workspace
- administrators will have authority to create folders in the private workspaces for which rules allow them access based on established project-to-user relationships
- user's will be restricted to folder creation within their private workspace areas only.

3.2.2 Inserting data objects into a configuration

Once a folder is created in DM2, items may be inserted into it through the use of the drag-and-drop capability of the GUI. Updates will be allowed to transient level folders only. The following steps/concepts will guide the DM2 implementation of insertion into configurations within the EFCM model--

- super users will be allowed to insert data into a folder regardless of its associated private workspace location
- administrators will be allowed to update folders residing in private workspaces for which rules allow them access
- users will be restricted to updating folders within their private workspace areas only
- working and released folders may not be updated.

3.2.3 Check out

DM2 provides a *check out* option that creates a new copy of the selected working folder, and allows the new copy to be modified. The original folder is superseded. File system items attached to the folder being checked out are copied to the current workspace with the same relative location as the original folder. A transient folder must first be checked in before it may be checked out. Thus a folder will be at the working version level prior to check out. The following steps/concepts will guide the DM2 implementation of checking out a configuration within the EFCM model--

- super users will be allowed to check out any folder from the shared workspace it resides in

- other users will be allowed to check out folders residing in shared workspaces for which rules allow them access
- baselined folders may not be checked out.

3.2.4 Check in

DM2 contains a *check in* feature which allows a folder to be returned to a shared workspace (vault), or transferred to a vault for the first time. If transferred for the first time, the folder becomes visible to the shared workspace and all it's children. A check in will not replace a predecessor of the same folder. Once a folder is at a mature state and is ready to be released, it may be "promoted" to the global workspace by using the DM2 *baseline* feature. All items attached to the folder are baselined as well. Changes are made to a released folder through the use of the *revise* feature. This feature creates the next revision of the released folder. The following steps/concepts will guide the DM2 implementation of checking in a configuration within the EFCM model--

- super users will have the ability to check in, baseline, and revise folders regardless of that folder's private/shared workspace location
- administrators will be allowed to check in, baseline, and revise folders residing in shared workspaces for which rules allow them access
- users will be allowed to check in/transfer ownership of a folder to shared workspaces for which rules allow them access
- baseline will work only on folders which are at the working version level
- revise will work only on released folders.

3.2.5 Accessing child versions

The DM2 expand relationship feature may be used to show a folder's relationships to other objects. DM2 allows relationships between objects to be displayed in a query window by selecting a folder and choosing from the available options under the info menu. If desired, DM2 can also provide a "tree" like view of these relationships. Child versions of a folder could be obtained by expanding the "is superseded by" relationship. The following steps/concepts will guide the DM2 implementation of accessing child versions of a configuration within the EFCM model--

- super users will have the ability to examine the relationships of any folder residing in any workspace location
- administrators and users will be allowed to examine relationships of folders residing in workspaces for which rules allow them access.

3.2.6 Accessing parent versions

The DM2 expand relationship feature may be used to show a folder's relationships to other objects. DM2 allows relationships between objects to be

displayed in a query window by selecting a folder and choosing from the available options under the info menu. If desired, DM2 can also provide a “tree” like view of these relationships. Parent versions of a folder could be obtained by expanding the “supersedes” relationship. The following steps/concepts will guide the DM2 implementation of accessing parent versions of a configuration within the EFCM model--

- super users will have the ability to examine the relationships of any folder residing in any workspace location
- administrators and users will be allowed to examine relationships of folders residing in workspaces for which rules allow them access.

3.2.7 Naming versions

DM2 will not allow the name attribute of a folder to be directly changed. Renaming a folder can be indirectly accomplished by using the DM2 copy feature. Copying the folder will produce a new folder object containing the same attributes as the source folder, but with a different name. The following steps/concepts will guide the DM2 implementation of naming versions of a configuration within the EFCM model--

- super users will have the ability to copy any folder
- administrators and users will have copy privileges for only those folders residing in workspaces for which rules allow them access.

3.2.8 Retrieving a named version

The DM2 *query* feature may be used to access the named version of a given folder based upon attribute values in the search criteria. The following steps/concepts will guide the DM2 implementation of retrieving named versions of a configuration within the EFCM model--

- super users will have the ability to query for folders residing in any workspace location
- administrators and users will be allowed to query for folders residing in workspaces for which rules allow them access.

References

[Cattell, 1991] Cattell, R.G.G., Object Data Management, Massachusetts: Addison Wesley, 1991.
[Martin Marietta, 1994] Martin Marietta RASSP Team, Martin Marietta Laboratories, “The Configuration Management Model for the RASSP System”, Moorestown, New Jersey, 1994.