# GME Limitations

## Introduction

I have been using GME to draw a variety of diagrams for UMLX and GMT and found it very useful, but of course there are areas were it could be better. This document is a summary of problems and WIBNIFs. It would be great if some of them could be resolved, however most can be worked around, so this document may just act as a repository for known limitations. Maybe they can be borne in mind for the reworking of GME under Eclipse.

## Tool facilities

### Bugs

The default aspect seems to be accidental. I think it may be inverse creation order of aspects in the chronologically first created paradigm sheet. But that's difficult to remember so tweaking XML is necessary.

ConnectionProxy cannot be used at the end of an AssociationClass.

AspectProxy is perverse. The Aspect should be defined once then proxied elsewhere for additional containment. There would then be no need for Equivalence.

Cut and paste fails on annotations. Annotations don't synchronize.

Edits of names, particularly new entries, in the Browser abort all too easily.

Auto-positioning has many quirks.

### Decorators

The Visual C++ support for decorators makes it fairly easy to create custom decorators pending a more meta approach possibly using a meta-icon paradigm. However only 'boxes' can be drawn by invocation of `Decorator::Draw(DC)` with the `DC` carrying the position and scaling. An additional entry taking a set of polyline nodes is required to support drawing of 'lines'.

### Libraries

Attaching a library is handy for providing the reference schema for texture mappings. However refreshing a library appears dangerous. So I'm very reluctant to do it, because I sometimes lose all the target information for ClassCopy.

### Upgrading

While developing paradigms an upgrade is regularly necessary, and for simple changes this often works well, although it would be nice to have to say Ok a little less often.

However when it fails diagnosis is poor, and not all upgrades are ok.

I once got a helpful, one-off opportunity to rename – nice, but it would be nicer if this supported renaming of Connection too. I suspect that a mapping should be built between each unidentified source identifier and an interactive choice of all available identifiers from compatible namespaces.

Introducing an abstract FCO inheritance for a Connection and then making the Connection visible via a Proxy of the FCO seems a definite problem. The Connection is not visible.

One common bad message is illegal object and a hex number. This can variously be a missing library, or mis-installed decorator. The former is perhaps a fragility of the authors access path in a subsequent recipients environment – needs an environment variable? I eventually discovered that if you succeed in clicking on Ok to update paradigm in the one second before the bad object error occurs all is ok.

It's difficult to be absolutely sure about some of these problems. I had far more, but eventually realized I had been fooled by Aspect re-ordering. When starting up a test usage of the updated paradigm, a change in the order in which aspects appear can make it look like things have collapsed when quite possibly nothing is wrong.

### Namespaces

I do not understand what the namespaces in GME are. It is certainly possible to have multiple same named Aspects, hence the need for Equivalence, however it doesn't seem to be as simple as regarding each as being in a namespace defined by its Folder and Model hierarchy, in some cases particularly with Attributes names must be distinct.

Requiring distinct names is pretty awkward, since the standard Meta paradigm does not support AttributeProxy or addition of Attribute to a Proxy, which requires that all re-use of attribute names must occur in the same diagram. This would be unacceptable for a fully fledged UML paradigm, where users must be free to add whatever attributes they like wherever they like. The above problem was relatively easily fixed by supporting AttributeCopy consistently in UMLX.

However a U2P practice whereby UML constructs grow as more sophisticated packages are used means that there is extensive use of inheritance from same named base. For instance, if a meta-model requires modeled expressions as multiplicity bounds, the Core.Abstraction.MultiplicityExpressions package is required, and it causes a bigger and better MultiplicityElement to inherit from the simpler MultiplicityElement that supports only textual values as multiplicity bounds. I found that subsequent attempts to create a ClassCopy of the more derived MultiplicityElement could misbehave by changing to null.

## Paradigm facilities

### The meta paradigm

Does not support AttributeCopy.

Does not support Attribute with a named type.

### The Example UML schema

The GME UML paradigm currently lacks support for the following concepts used in class diagrams in the U2P proposal.

Lines cannot be visibly named, or marked as derived.

Line ends can only have roles and multiplicities. [Workaround is to add {subset} as tail text to role, and {union} as tail text to cardinality.]

Line ends cannot have navigation arrows. [NavigableAssociation added in UMLX]

Shared aggregation cannot be shown. [Not used in U2P]

Aggregation parents cannot have multiplicity. [Added to UMLX]

Atrributes cannot have arbitrary types. [TypedAttribute added in UMLX]

Attributes cannot have initialisers. [Allowed in UMLX]

Operations cannot be specified. [Abused attribute text in UMLX]

ClassCopy needs to show (from xxx). [pending as a Decorator upgrade for UMLX]

(enumerated) would be nice

Attribute, role, derivation, multiplicity is not explicit [Informal extra text in UMLX]

Attribute cannot be added to a ClassCopy [Allowed in UMLX]

**Ergonomic limitations/bugs**

Lines cannot be dragged from one node to another.

The hover text doesn't indicate the defining diagram to resolve multiples.

Picking a specific edge of a Connector can be really hard even when zoomed large.

Elements cannot be reclassed, e.g. to change one kind of connection to another.

An AttributeProxy is needed to avoid putting all uses of same attribute in same picture.

Visualisation is not supported from derivation via a Proxy.

Refresh Library can lose all Proxy bindings.

## Required Facilities

UMLX mapping diagrams comprise correspondence arrows between schema sub-graphs. It would be nice if these could be drawn more directly.

The nodes of a sub-graph can be selected and copied individually but not en masse. It is necessary to copy copies, instantiate copies of definitions, redraw lines. Too much is not copyable – some could be fixed e.g. InheritanceCopy, but I suspect that ConnectionCopy requires internal support since lines are less meta than boxes in GME.

It should be possible to Control-Shift drag a group (set?) of boxes and lines. This would ensure accuracy and could enable rapid determination of the impact of changes to a schema as regards all textures that source or sink the changed region.

It would be nice to be able to print multiple diagrams more easily so that they can be delivered with editorial content. Ideally there would be full OLE automation, so that they could be embedded or linked from Word. As a work around it would be adequate to print auto-clipped diagrams to file names matching the diagram and folder naming. At present using flash.exe from the free download Microsoft HTML Help Workshop provides a useful way of capturing and clipping an inner frame. I find that PNG format is good quality and compact.

It would be nice to have File->Close All Models, to prepare for opening a couple and tiling them for copying. Even better, allow a multi-selection in the browser to be explicitly tiled.