



ALLEGRIA SOFTWARE, INC.

A Technical White Paper

by Ruiwei Wang

Sr. Software Engineer

November, 1996

The Integration of ForReview with Metaphase

Overview

Metaphase series 2 is an enterprise-level product data management (PDM) system. Metaphase Series 2 software manages a company's product data and product development processes from initial concept through manufacturing and delivery. Tracking such processes in a typical environment requires that thousands or hundreds of thousands of images, technical drawings, and documents be stored in the PDM's central repository. Moreover, the exchanging of this data between co-workers and vendors can generate massive amounts of comments and change requests that also must be stored and tracked.

To address this need, **Allegría Software, Inc.**, an innovator in imaging, documentation and graphics applications, recently integrated its enterprise-wide view and mark-up product, *ForReview for Windows*, with *Metaphase Series 2*. ForReview's very large and broad list of supported file formats, its powerful mark-up support, and its ability to be customized differently among the various users within an organization, makes it the perfect choice for companies that have adopted Metaphase. Because ForReview can be customized for each "type" of user involved in the review process, — administrators, designers, engineers, reviewers, assembly personnel and clerks, — only one viewing application need be purchased and deployed within an organization, obviously reducing installation, upgrade, training, and support costs. More importantly, mark-ups made on any repository object can be made available to any user. Since virtually all file formats and both PC and UNIX platforms are supported, there is no need to deploy costly CAD applications to every desktop.

This integration, called ForReview for Metaphase, consists of several components that are seamlessly integrated into the Metaphase system using the Metaphase Integrator Toolkit and the Application Programming Interface (API) provided by ForReview. The Metaphase system is object-oriented, having a three-tier client server architecture. The client communicates with a set of dedicated servers to execute an object method, and a dispatcher is used to manage the servers and route the requests on the network. Consequently, the integration is done on both the client and the server side. These modules communicate with ForReview and Metaphase via network Dynamic Data Exchange (DDE) or through OLE Automation. It should be emphasized that the underlying code of ForReview is not changed in any way.

The ForReview for Metaphase integration adds a set of menus to the *Action* pull-down menu in the Metaphase client, and is available in various object browser windows. These menus, *View*, *Markup*, and *Print*, allow the user to quickly access these three ForReview tasks. ForReview always protects original documents by not allowing them to be edited in any way. Marking-up is always performed on user defined layers, and saved in a separate file called "mark-up" file (recognized by the *.mkp* extension). To accommodate these mark-up files in Metaphase, a new object class, *Markup*, is added to the system class hierarchy. Objects of this type can be instantiated by registering ForReview-native mark-up files with the Metaphase system. The user can create various relationships between a *Markup Object* and a document object in the usual manner. These relationships, "*Has Markup*" and "*Is Markup Of*" can be multi-lateral, and can be manipulated using existing Metaphase facilities.

The integration takes advantage of the Metaphase system security control by requiring the user to manually check out the mark-up(s) for modification, and to check them in when done. This is a convention rather than a limitation. In the case of a new mark-up file created in ForReview, the integration does the necessary work so that the new *Markup Object* can be later checked in properly by the user into the Metaphase system.

Functional Architecture

At the server level, two classes, *ForReview* (parent *Tool*) and *Markup* (parent *NoEdText*), have been added to the Metaphase system class hierarchy, and each has messages defined for it in terms of the Metaphase MODEL language. An external server, *ASISVR*, is created on the server side, which provides services for ForReview-related requests from a client. At the client level, a proxy application (in the form of an EXE) and a custom client server (in the form of a DLL) is created. The EXE is the one who actually launches the ForReview application and does the necessary housekeeping for data received from the server before passing them to ForReview. The custom client

is needed when a new mark-up file is created in ForReview and a new *Markup Object* and the corresponding relationship with the base document has to be created in the Metaphase system. The functional architecture is illustrated schematically in Fig. 1.

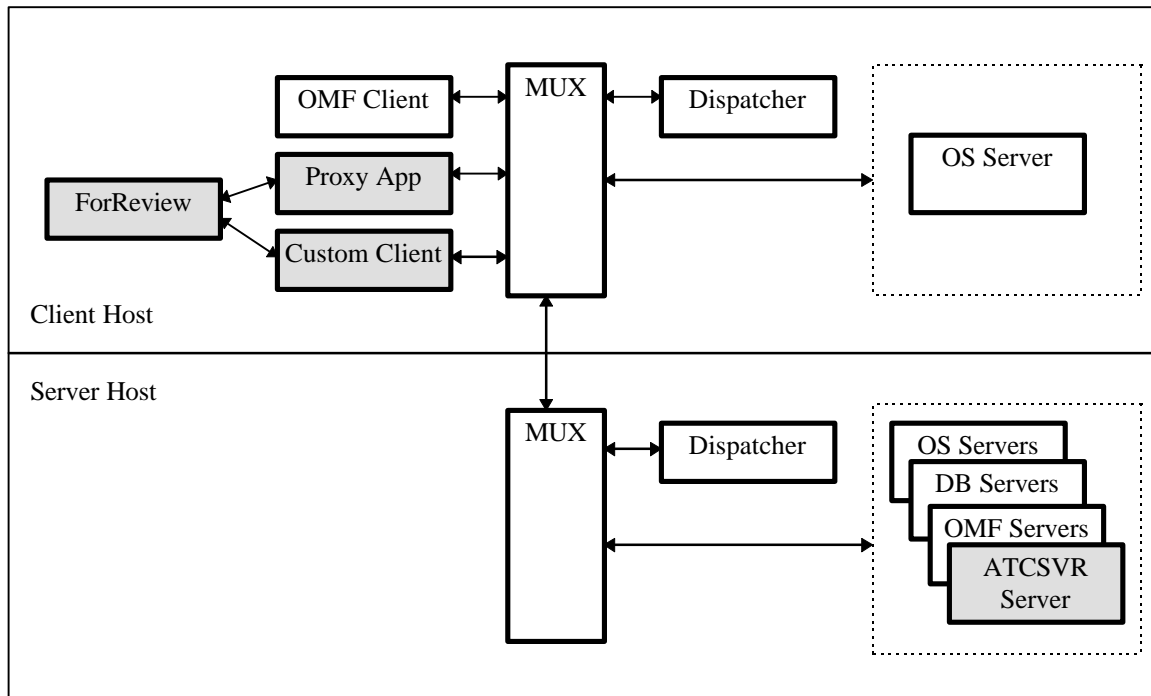


Fig. 1 Functional architecture of ForReview for Metaphase

It is clear that ForReview for Metaphase allows ForReview to be tightly integrated with Metaphase, not just to be “launched”. When working in Metaphase, ForReview is started only when a ForReview function is invoked. In the event that the connection between Metaphase and ForReview is lost (due to a network failure for example), any work done in ForReview can be saved locally, and the mark-up files can be registered with the Metaphase system after the connection is re-established. This offers an additional layer of protection for the users work. At any time, with or without the integration, the user can use ForReview to view and mark-up files that are local or outside the Metaphase database, but the user will obviously not be able to access any files stored in the Metaphase system if the user is using ForReview independently.

To modify or create mark-ups, select the base document to be marked-up from the appropriate Metaphase browser window. Next, select the *Markup* option from the *Action* pull-down menu, or double-click on the document (if *Markup* is the default double-clicking behavior). At this time, the ASISVR server is notified of the event, and the appropriate method is called with actions delegated to the proxy application. Specifically, a pointer to the document object is retrieved from the Metaphase system, and a copy of the object content is made and cached in a local repository on the user’s host. Next, a list of *Markup Objects* attached to this document is collected into a Metaphase-native browser dialog, allowing the user to select which ones are to be loaded into ForReview. Notice again, that all mark-up files intended to be changed must be manually checked out to the user’s local host at this stage, and the integration delegates security control to the Metaphase system proper. After the user makes the selection, control flow will be relayed to ForReview.

Users can now proceed to redline or “mark-up” the base document using facilities provided in ForReview. ForReview for Metaphase keeps track of the base document and its mark-ups as “external”, to distinguish from “native” mark-ups not from the Metaphase system. When the user has finished marking-up the base document, the integration modules are notified of this event through callback functions set in ForReview either by closing the base document or by explicitly unloading a particular mark-up file.. For new mark-up files, the custom client will

then be invoked, sending out requests to the ASISVR to register the new *Markup Object* and establish the bi-directional “*Has Markup*” and “*Is Markup Of*” relationships with the base document. These relationships are visible through the usual right mouse button action. As noted above, the user is responsible for checking in all *Markup Objects* that have previously been checked out into the Metaphase system.

Modules in Action

Following is a demonstration of the integration modules in action. There are no menu changes to ForReview when running ForReview for Metaphase. However, ForReview for Metaphase adds three menu entries, *View*, *Markup*, and *Print*, under the *Action* pull-down menu of the OMF client, allowing the user to quickly access these three ForReview tasks (Fig. 2).

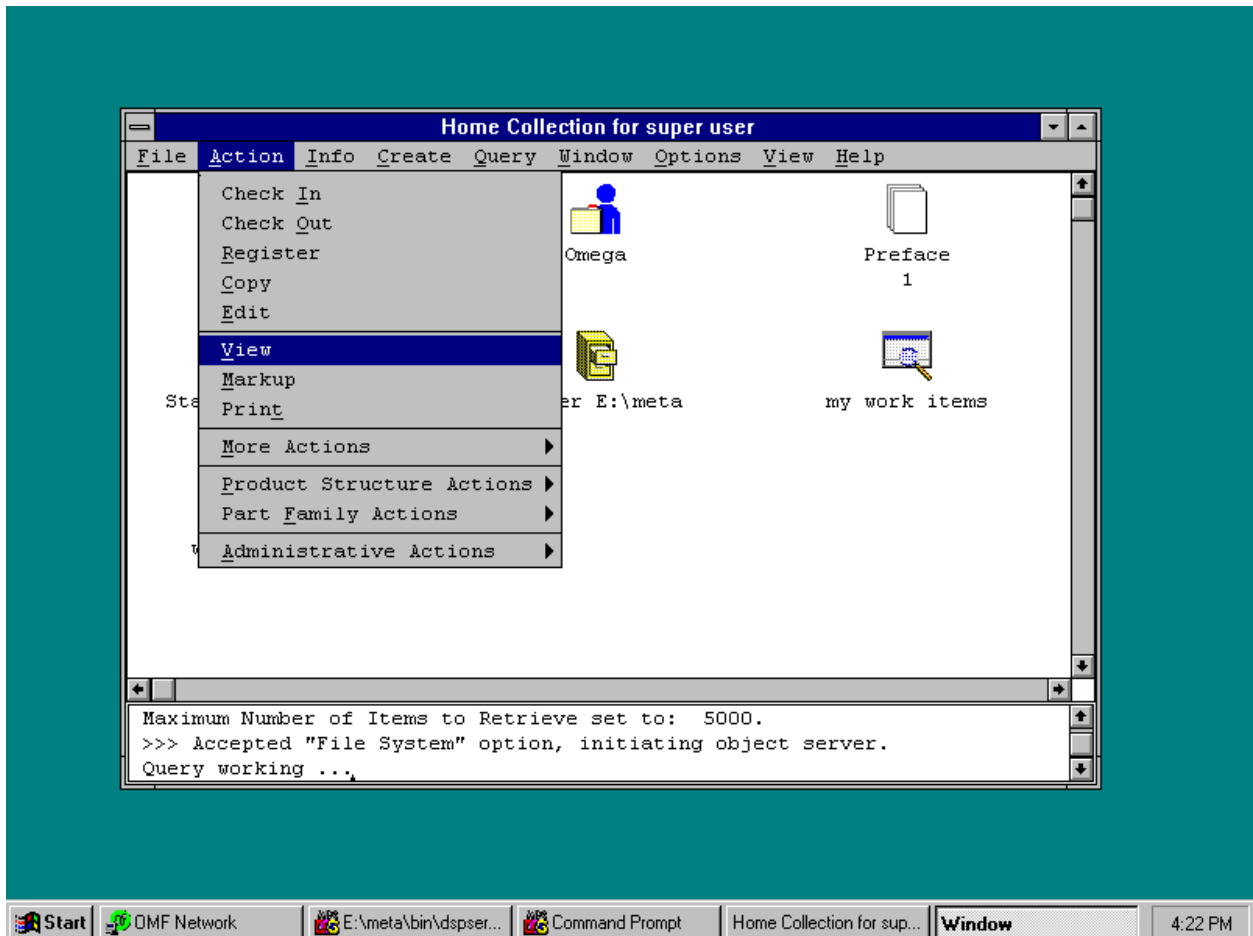


Fig. 2 Menu changes to OMF client

To mark-up a base document, (in this case, *bike.cgm*) highlight the base document and select **Markup** from the **Action** menu.

ForReview is automatically launched, if it is not already running, and the base document, (*bike.cgm*) is loaded. The **Select Markup Files** browser dialog appears allowing the user to choose what associated mark-up files are to be loaded with the document. The user can utilize the existing Metaphase facilities to browse through the attributes of a *Markup Object*, further aiding him or her in selecting which mark-up to load into ForReview.

After the browser dialog is closed, the selected mark-ups will be loaded. The user can proceed to “mark-up” the base document in the usual manner, using facilities provided in ForReview. When finished marking-up the base document, if a new mark-up file is created, the user will be prompted with a Metaphase-native *Attributes* dialog to set any relevant attributes of the mark-up file. The user can now check in the appropriate *Markup Objects* back to the Metaphase system.

This completes the steps involved in a *Markup* operation. The two other modes of operation, being *View* and *Print*, operate essentially in the same manner. That is, the user is not allowed to make any modifications to the mark-ups. Note that the *Print* operation will allow the user to print directly from within *Metaphase* using *ForReview*. These three modes are the basic operations that can be performed on a base document and its mark-ups.

Conclusion

In conclusion, we have demonstrated that ForReview, a state-of-the-art view and mark-up tool has been tightly integrated with Metaphase, an enterprise-level product data management system. We believe this challenging integration has met the typical industrial need in viewing and managing images, technical drawings and office documents in a demanding way. This integration can also serve as a template architecture to be followed by other integration works.