The creation and sending of an enhanced e-mail message according to the present invention, as well as initial receipt of an enhanced e-mail message prior to a user event, will now be described with respect to Figures 10 and 11.

FIGS. 10-10B illustrate the process of creating and transmitting an enhanced e-mail message with a container attachment that is self opening and has the other characteristics described herein. In step 62, header information is first designated to define the container. Thereafter, component definitions are created by the sender in a step 64. There can be any number of component definitions desired, with it being understood that the more definitions that exist, the larger the size of the attachment that will ultimately be transmitted. Thereafter, sub-component definitions can be created for each component object, as described previously, is step 66. In step 68, the component and sub-component files are combined into an ActiveView file, which thereafter is combined with header information and attachments to create a container file in step 70. The container file is encrypted in step 72. Once being encrypted, in the preferred embodiment, the Java class files, collectively termed the recipient executable software, and the native stub are attached to the encrypted container file in step 78. Mime encoding follows in step 80, which mime encoded file is then attached to an e-mail message in step 80 and transmitted.

Upon receipt at the recipient computer in step 84 with reference to FIG. 11, double clicking the received message in step 86 if on a java virtual machine will cause self opening to occurs in step 88, or, alternatively, opening the container attachment. Thereafter, the files are extracted from the attachment in step 90 and invocation of the application (the java virtual machine and the java class files) and extraction of the senders container data files in step 94 will cause the display of the container in step 96. Thereafter, in step 98, the computer 30 awaits the recipients action, as will be further described hereinafter.

As a result, of the display on the display 38 of the recipient computer 30, a container, such as the envelope container 100 illustrated in FIG. 12 is obtained. At this time, the recipient executable software then proceeds to a state in which it waits for the user to take an action, as will now be described with respect to FIG. 13.

In step 110, a first event takes place. Step 112 follows and a determination is made in step 112 as to whether the event that takes place is an event that is directed to one or more object components. If it is not such an event, then step 114 follows and the recipient executable software does nothing, and waits for a next event in step 120, Types of nothing events would be, for instance, if the mouse cursor 122 illustrated in FIG 12. moves to an area that is not over any object component.