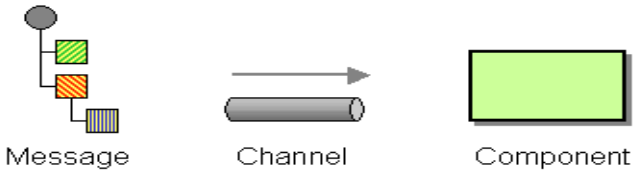
DIAGRAM NOTATION

Integration solutions consist of many different pieces—applications, databases, endpoints, channels, messages, routers, etc. If we want to describe an integration solution, we need to define a notation that accommodates all these different components. To our knowledge, there is no widely used, comprehensive notation that is geared towards the description of all aspects of an integration solution. The Unified Modeling Language (UML) does a fine job of describing object-oriented systems with class and interaction diagrams, but it does not contain semantics to describe messaging solutions. The UML Profile for EAI [UMLEAI] enriches the semantics of collaboration diagrams to describe message flows between components. This notation is very useful as a precise visual description of a system that can serve as the basis for code generation as part of a model-driven architecture (MDA). We decided not to adopt this notation for two reasons. First, the UML Profile does not capture all the patterns described in our pattern language. Second, we were not looking to create a precise visual specification, but images that have a certain 'sketch' quality to them. We wanted pictures that are able to convey the essence of a pattern to the reader at a quick glance—very much like Alexander's *sketch*. That's why we decided to create our own 'notation'.

Luckily, unlike the more formal notation, ours does not require you to read a large manual. A simple picture should suffice:



Visual Notation for Messaging Solutions

This simple picture shows a message being sent to a component over a channel. We use the word *component* very loosely here—it can indicate an application that is being integrated, an intermediary that transforms or routes the message between applications, or a specific part of an application. Sometimes, we also depict a channel as a three-dimensional pipe if we want to highlight the channel itself. Often times we are more interested in the components and draw the channels as simple lines with arrow heads. The two notations are equivalent. We depict the message as a small tree with a round root and nested, square elements. The tree elements can be shaded or colored to highlight their usage in a particular pattern. Many messaging systems allow messages to contain tree-like data structures, for example XML documents. Also, depicting messages in this way allows us to provide a quick visual description of transformation patterns—it will be easy to show a pattern that adds, re-arranges or removes fields from the message.

Source: http://www.enterpriseintegrationpatterns.com/patterns/messaging/Introduction.html