A reference model. Presentation, abstraction and control (PAC) is an implementation model that attempts to bridge the gap between the application and the user interface. The Presentation defines the input and output behaviour, the Abstraction implements the functionality of the underlying application and the Control maintains the mapping and the consistency between the abstractions and the presentations.

An extended model. Command, Arguments, Objects and Selectors (CAOS) extends the PAC model by modelling more of the structure of the application. In PAC the model of the application, the abstraction, only consists of objects. CAOS contributes support for navigation, invocation of methods and selection of arguments as part of the model. The control part of PAC is in CAOS handled by attaching general control objects to the presentation. No specific code programming is then needed for the control. There are four basic kinds of control objects: Commands on objects, Arguments to such commands, Objects and Selectors of values for arguments. Each of these objects represent some entity from the abstraction, and connects it to the presentation. Control objects also store navigation information, such as the name of a method or an argument. This information is used together with the parent control objects to find the abstraction entity that should be represented. The presentation part in CAOS consists of a tree of interaction objects augmented with navigation specifications and CAOS state objects.

This example shows how a layout might look on the screen. The visible parts are the first and last name of a person, along with a text input field for a new telephone number.

This is the tree structure for the example. The presentation tree is shown to the left, the tree of control objects to the right. The strings in parentheses are the navigation specifications. The grey lines indicate the connections between the trees. An important goal of the model-based tools is to be able to model any kind of user interface. This means that the modelling language must be very expressive. The CAOS approach is instead to create a simple system, by restricting the domain of interfaces addressed and building design decisions into the system.

The language. View Definition Language (VDL) is a language specifying user interfaces. The basic approach of the language is to specify attribute/value pairs for a hierarchy of user interface objects. The language is declarative and puts an emphasis on expressive power through the use of inheritance, parameters and variables. In VDL, the description of the sets of attribute/value pairs for the complete hierarchy of components forming a user interface is broken up into templates, or styles, describing a part of the hierarchy. A definition of such a style consists of a name, a parameter list and a sequence of definition items. The example style foo is defined to take one parameter, bg, and uses it for the value of background. The style bar inherits from foo, passing a parameter of blue. The result is that bar gets a background of blue, and a foreground of red.

CAOS and VDL
Mats Johnson, Erik Borålv, Bengt Göransson
http://www.cmd.uu.se/ikaros/