Simulation and Modeling with Ideal Circuit Elements

One method of determining how a collection of "real" components behave together in a system without actually building the system is known as *simulation*. Simulation allows us to virtually construct a circuit an determine its behavior, typically prior to construction. This is an advantage when:

-components are not available for building a prototype
-components may be destroyed as part of the experiment
-the experiement is dangerous to do (lighting striking a high voltage line)
-the cost of building the prototype is very high (i.e: building a Pentium 4)

"Real" components in a system may exhibit very complex behaviors. However, often many behaviors will not be relevant to the solution, or they may effect the results only slightly. Therefore, to simplify simulation of real-world systems, we replace the real components with *ideal models* of the real components. These ideal models only have the relevant behaviors required to obtain accurate simulation results. The simplified ideal models are more easily understood and also make the simulation run quickly.

Our assumption of wires as perfect conductors is an example of using a model. Wires *do* have resistance, but for the magnitude of resistance and current involved in most of our circuits, the assumption of perfect conduction would not alter our answers even 1%.

Depending upon the results needed, and the application context, sensitivities to the *properties* of the model may change. For example, say we were looking at the voltage available to a automobile starter motor at engine cranking. When cranking, approximately 250 amps are drawn. Resistance in the starter wiring as small as 0.01 ohms will cause an error of 2.5V or over 20% in a 12V system. An off-hand assumption of perfect (no resistance) starter cables would cause excessive errors.

As long as we understand the differences between ideal models and real-world components, and the context of application, we can replace complex components with ideal components and make problem solving much easier. Just keep in mind the limits of your chosen model and of the context in which its used.