

Validation is the process of evaluating input and judging it valid or invalid. Such evaluation subjects the input to a battery of “validation rules” that evaluate the input in the appropriate context. For example, if the user enters a “committed delivery date” we might want to ensure that:

- The committed delivery date is reasonable in the abstract, e.g., occurs in the future
- It is possible to deliver on that date given the availability of the desired products, and the currently selected shipping method, and whether there is enough time to prepare the goods for shipping
- The order is “shippable”, e.g. the customer’s credit has been verified, the address is legitimate, and the total is within the limits authorized for this user

Clearly such rules can be complex, involving not only the input value itself but also the state of the target object (the order), facts about related objects (customer, shipper, product), and aspects of the environment during the validation (time of day, the user’s role).

User input validation gets most of the attention but we need to validate programmatic inputs as well. That delivery date could as easily be set by business logic or a web service request as it is by a wayward click on a calendar control. The rules are the same for everyone, human and machine.

Validation is hard to do well especially as the application grows and validation criteria change. Common failings include:

- Missing and incorrect validity checks
- Inconsistent checking
- Failure to validate at the right times
- Poor communication with end-users
- Inadequate mechanisms for correcting mistakes

Enterprise application developers are looking for a robust validation system that operates consistently and reliably across a large application. Robust validation cuts both “vertically” and “horizontally”:

- We validate “vertically” when we validate several times in multiple layers of the application. We want to validate in the client UI layer so we can give immediate feedback to the user. We may need to validate again when we save, even though the objects we save are no longer on screen. We may even need to validate again on the server side to protect against misadventure coming from outside the relative safety of the hosted environment.
- We validate “horizontally” when we apply the same mechanisms uniformly across all modules of the application. If the user can set the delivery date on any of several screens, the same rules ought to apply – unless, of course, there is something special about a particular screen.