



By [Heidi Adkisson](#)

Objects and actions analysis is a method of documenting what data (objects) need to be manipulated and what functions (actions) can be performed on the objects.

A key benefit of the analysis is representing system functionality without consideration of how the interface needs to look or behave. At Blink we have found that conducting objects and actions analysis at the beginning of the project helps us to:

- Understand the system's breadth of functionality in a compact, "light weight" representation
- Develop a concise, consistent vocabulary for the system
- Ensure all system tasks are accounted for in early-stage design work
- Simplify the user interface by making clear, explicit associations of objects and related actions

To understand the idea of objects and actions, let's look at a simplified matrix - one for an e-commerce site that sells consumer electronics. Basic objects for this system are a customer account, a product, and an order. Some basic actions are create, edit, send to a friend, and purchase.

Objects are listed down the first column; actions are listed across the first row. Note that products and orders are not user-created objects. These are created by behind-the-scenes processes (products may be uploaded via an automated process; the system creates orders a result of a purchase).

For this e-commerce site, there are some additional objects to consider. A product can have one or more reviews. A product also has an associated product manual which users can download before purchase.

Reviews and the product manual are child objects of a product, meaning they are only found in association with a product. As such, we show these under the parent product - indented and in italics.

Users can also create certain types of lists (or aggregates) of products. For example, they can add products to a cart or wish list. We show list-type objects also under the Product object, but using brackets to indicate them as aggregates.

Most systems we work on at Blink are much more complicated. Additionally, we often work from a mix of requirements sources - formal (but perhaps outdated) written requirements, existing functionality, and requirements communicated verbally. Creating an objects and actions matrix knits these sources together into a single, concise view of required functionality.

The list of actions in particular can be helpful in making sure system vocabulary is consistent. By showing actions across all objects inconsistencies in terminology can be easily identified.

Objects and action analysis can also tease out holes in requirements. Going through the matrix systematically examining which actions apply to which objects can identify additional, helpful functionality.

We like to do objects and actions analysis concurrently with development of scenarios and behavioral profiles. Both perspectives compliment each other: scenarios takes a more detailed look at key functions and the likely sequence in which they are performed while object and actions looks at the system from the broader perspective. Both perspectives are important when moving into development of the user interface.

Finally, the matrix view helps [organize elements in the user interface](#). Our experience with usability studies has shown that for web-based systems, users expect all actions for an object to be in close association with that object. For Windows or Macintosh client-based applications, objects and actions analysis helps arrange commands on the screen and organize the menu structures.

The goal of the objects and actions analysis is not to create the perfect model, which can result in analysis paralysis. The goal is to create a model that hangs together logically - one that will allow you to examine the system as a whole before jumping into the design. In our experience, this has an excellent payoff in clarifying requirements and in starting design work with a clearer vision of what needs to be developed.

[Contact us today](#)