Table of Contents

Usage Scenario

A *usage scenario* is a functional description of how a product might be used by an external agent, used to analyze designs.

What is a usage scenario?

A way to describe the steps to achieve a goal

A *usage scenario* (US) is a step-by-step description of one way in which an intervention might be used. The US describes the steps that an external agent might take to achieve a goal using it; they represent how a product is *intended* to be used.

Usage scenarios typically cover three main stages of any part of a product's life be it manufacturing, use, maintenance, or end-of-life. Our focus in this design course will be on the *use* stage of the product lifecycle. For our purposes, the three stages covered by USs are:

1. **Initialization.** This involves all the steps needed to get a product ready for use.
2. **Use.** This involves the steps executed to actually use the product.
3. **Finalization.** This involves all the steps needed to clean/store/put-away the product.

Usage scenarios are annotated diagrams

A US can take many forms. In engineering, we prefer to use flowchart-like diagrams to describe the steps in the scenario. To prevent diagram clutter, if there is essential information beyond just naming a step, we tend to use footnotes, endnotes, or (in software) links to other documents. The US itself should focus on the overarching, high-level view of how different steps fit together into an overall process.

A US diagram **must** be annotated in particular ways to connect it to other parts of your project.

A properly annotated US includes the following.

1. Each box/node/step in the diagram must have a unique identifier of the form `US.S.N`, where
   - US is a unique numeric identifier for the US (starting at 1);
   - S is either 1 (for initialization), 2 (for usage), or 3 (for finalization); and
   - N is a unique preferably sequential integer identifying a specific step.
   - Examples:
     - `3.1.5` names the 5th step of the initialization stage of US 3, and
     - `1.2.14` names the 14th step of the usage stage of US 1.
   - In the US diagram itself, you can omit the US part of the ID to save space. However, when referring to a specific US step elsewhere in your report, you **must** include the US ID to distinguish it from other USs.
2. The three stages may appear together in a single figure. However, if the US gets large, it makes

sense to break the diagram into three figures, one for each stage.
3. Some optional text, written in extended point form, to highlight points of interest about key steps in the US.
    - You can use the numbering system of the steps in the US diagram to index a numbered list of key points.

Usage scenarios capture HMILs

Ideally, each step in a US represents nothing more than a single human-machine interaction loop. It may take multiple passes through the HMIL to achieve the goal of a given US step.

For example, consider the first step of the US shown in Figure 1 below: `Grasp blender`. During execution of this step, the following HMIL happens:

1. the user observes the blender,
2. the user determines (cognitively) how and where to grasp it,
3. the user moves so as to position their hands and actually grasp it; then
4. the blender experiences pressure exerted by the user's hands,
5. the blender deforms (hopefully only ever so slightly) in response to the user's hand pressure, and
6. the blender provides resistance and haptic response to the user, informing the user that they have made contact with the blender.
7. the user realizes (cognitively) that their grasp is not good.
8. loop to step 3.

This is just what an HMIL is.

If you cannot imagine a HMIL loop for each step in a US, then you're doing something wrong.

Usage scenarios are functional

Since we do not know (yet) exactly what the intervention will look like, we must keep the usage scenarios *functional* in nature; that is, **usage scenarios must make no commitment to structural aspects of the intervention**.

Exercise for the Reader

Consider the differences between an old personal digital assistant[1)] and a typical agenda. While they are completely different structurally, they are *functionally identical*, and so would have the same basic USs.

Sketch out the US for typical usage for both a PDA and an agenda to prove it to yourself.

USs are context specific, and so are typically framed with respect to one or more Personas and a SUC, Despite their functional nature, USs must be quite specific and treat specific activities and tasks, in an “intended” way in which the intervention ought to be used.

Think of a US as a flowchart version of an operating manual that doesn't refer to product's parts but only to the functions they execute.

Since you are working on a specific concept, you may well have made commitments in the concept to specific structures. For instance, if you're designing *a way to blend foods*, your concept may include buttons to select speeds, etc. However, you may find that you need to replace the buttons with a dial.

- To avoid unnecessary edits to the USs, it's best to keep them functional and not to refer to specific embodiments.
- In this example, `select speed` is a better way to describe a task in a US than `push the required speed button`.

The table below contains other examples of how to avoid structural statements in USs.

Table 1: Examples of structural vs functional US steps.

STRUCTURAL DESCRIPTOR	**FUNCTIONAL DESCRIPTOR**
Push the elevator call button.	Request an elevator car.
Turn the blender's speed control to the desired setting.	Set the speed of the blender.
Lift the car's hood.	Access the car's power system.
Turn the deadbolt.	Lock the access system.

Usage scenarios give operational overview

There are many different USs for a given design intervention. For instance, consider a photocopier. There could be several possible USs just to cover how end users may use the machine:

- making copies of documents;
- scanning documents to PDF format rather than to paper copies;
- refilling the paper trays;
- and so on.

Then there are USs for users who maintain the photocopier: clearing a jam; replacing toners and fixers; replacing broken rollers; and so on. Some of these tasks might be performed by “users” but others will only be done by qualified maintenance personnel. Even just a moderately complex design could involve dozens, or hundreds, or even thousands of USs.

In this course, you only need demonstrate sufficient mastery of the basics of US construction and specification, so make sure you prioritize quality of the US over its breadth and scope.

This also speaks to the need to keep USs well-organized.

- In the example above, there might be many steps common to multiple USs.
- Depending on the complexity of the intervention, one might have to have separate USs for the common steps to avoid potentially disastrous duplication of information.
- One can use typical flowcharting conventions to show where two different USs connect to one

another.

Usage scenarios identify failures that lead to interaction errors

A US represents all the steps needed for an intervention to help a user reach a goal. Each step in a US represents one HMIL. Thus, a US is a comprehensive tool to look for interaction errors as well as product failures.

Once you have a good US, you can then focus on each step, thinking of it as an HMIL, and consider the many ways that the intervention could fail and/or that the intervention might mismatch the user's abilities and needs.

You can embed the most significant or likely failures and errors directly into your US to document them.

You may not be able to eliminate every potential product failure and interaction error, but you may be able to find ways to mitigate them - manage them in some reasonable way so that the failure or error does not lead to bad outcomes.

While no one can ever fully predict the ways in which a product can fail, it is a matter of *engineering professionalism* for designers to be *diligent* in seeking out potential failures and trying to mitigate them.

Students need not try to find every possible failure and error of an intervention. However, they must show that they are aware of the potential for these undesirable events, and able to qualitatively identify the most significant / most likely failures and errors that might occur.

How do you construct a usage scenario?

Given the design roadmap, a US is specified for a given design concept, a given Persona, and a given SUC.

This means there will be as many USs as there are Personas in a given team. All the USs will treat the one design concept, but from the *point of view* of individual Personas in their respective SUCs.

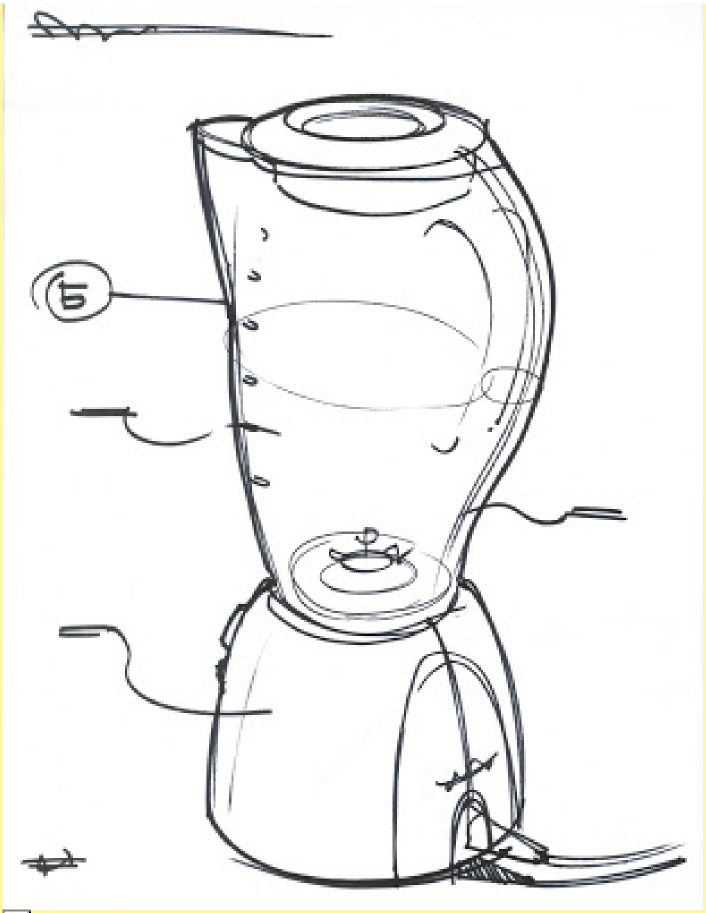
Usage scenario generation process

Here is a general procedure for developing usage scenarios in team projects. The use of extended point form is strongly encouraged.

1. **Sketch the US**
    - Each team member develops a US diagram (with annotations are required) for their own Persona and SUC.
    - This is best done with paper and pencil in your design journal so that you can “play around” with it to make sure you have all the necessary steps.
2. **Review and Integrate the US sketches in a team meeting**

  - In a team meeting, merge all the US diagrams into a single, generic baseline diagram that represents how the product you're designing would be used by a generic user. Think of this US as a *visual instruction manual* that you might ship with your product.

3. **Develop a full diagram**
    - Using an app like https://www.diagrams.net/, generate an “online” version of each US.
    - Make sure all the USs have a consistent appearance and structure to make them easier to understand to “outsiders”.
4. **Add annotations and descriptive text**
    - Make sure the actions are clear enough that a reader unacquainted with your project can still make sense of it.
    - The rule of thumb is that if you struggled to encode a particular task of sequence of tasks, then you can expect the reader to need a little extra explanation to understand it.
5. **Add interaction errors and product failures**
    - The number of failures and errors you add will depend on the size and complexity of the diagram.
        - Remember that the goal is to demonstrate you can make good decisions about product failures, not include every conceivable failure/error.
        - Examples of product failures: an elevator's motor burning out during operation; a wine glass becoming permanently stained from holding red wine.
        - Examples of interaction errors: a user being unable to tell if they have successfully instructed an elevator car what floor to go to; a wine glass slipping out of a user's hand.
        - Focus on the failures and errors that will most seriously affect the performance of your concept.
6. **Final review by the team**
    - Every team member should review every US diagram one last time to ensure accuracy, brevity, and completeness.



A “blender” concept sketch. (source)

Example: a way to blend foods

A particular team member has developed a concept as shown to the right. The team member's Persona is *Zoya*. Zoya is known to be elderly and has arthritis in her hands. The SUC is described below.

Table 2: One possible SUC for a *making smoothies*.

SUC 2	An elderly person makes smoothies for their grandchildren.
Owner	Archer
	On a warm summer afternoon, a grandparent makes smoothies for their four grandchildren, who are visiting. The grandparent is alone except for the children, and they are in the house belonging to the parents of one of the children. The product is stored on an upper kitchen cabinet.

One possible usage scenario is shown below.

Figure 1: One possible Usage Scenario diagram for the reference blender.

Some notes about this example are in order.

Regarding formatting:

- The typical US diagram is not linear (as is evident from the branches in Figure 1. There could be many, many branches. Try to capture the most reasonable ones you expect.
- Notice how the three stages (initialization, use, and finalization) are rendered graphically. You may use this layout, or some other layout of your choosing *so long as the three stages are clearly evident*.
- The use of colour separates the main flow of each stage from the product failures. You may use different colours, *so long as you use them consistently*.
- Interaction errors are **not** shown in the example. These are left as an exercise to the reader.
- Curved corners on arrow links are best. This is evident where links merge or split. Most software packages for diagramming support this.
- Notice how multiple links heading in the same direction overlap. This helps keep the diagram tidy. Most software packages for diagramming support this.
- Notice the numbering scheme of the nodes.
    - Flow nodes are numbered as described above.
    - Product failures are numbered consecutively from 1.
    - Actions that follow a product failure are not numbered.

Regarding diagram structure:

- Notice the level of detail of the main flows. We should be able to envision modelling each task with a HMIL. The product failure nodes are not as important in this regard; they can be represented in more general terms.
- The nodes marked `Go to Maintenance` indicate that there is some other US diagram, called *Maintenance* to which one would refer if those circumstances arise.
- No description is given of what happens *inside* the blender (e.g., a switch is closed, a motor accelerates to a predefined speed, etc.) These are behaviours that are unknown at this level of design. They will become evident later, but only *in support of* achieving the performances that users expect.

Regarding intended meaning communicated by the diagram:

- The product failures are *internal* to the product. We represent them only insofar as they result in degradation of product performance that the users would experience, possibly as interaction errors (which are **not** shown in this example).
- Steps that occur outside of direct interactions between the product and users are ignored.
    - For instance, `2.17` and `2.18` make no reference to what the user will be pouring *into*. We are not designing the item into which users pour, only the thing from which we pour.
    - This is not to say that the item into which a user would pour the carafe contents is entirely irrelevant. It is only irrelevant here because we would expect such information to have been provided as part of the context of the SUC or as a requirement. We will return to this point in later stages of the design process.
- Step `2.22` has manifestly different HF implications than `2.15`. We might have considered looping from the YES branch of `2.21` back to `2.15`; but this would be **incorrect** because the specifics of the actions a user would take to remove the carafe from the blender base are probably very

different from those of lifting the carafe from a table or countertop. These HF factors are essential to a good design and so you *must* attend to them in detail.
- Step `3.5` could well connect to an entirely different US, one that focuses on how exactly to clean the blender. Anyone who has had to clean the carafe base (the part that has the blades attached) knows how challenging that can be. Whether a given team does or does not add another US for operational aspects such as cleaning will depend on each team. Remember, it is better to provide one truly outstanding US than two bad ones.

Regarding specific product failures:

- `(1) Plug doesn't fit` is not the result of the user's attempt to plug the blender in, but could be for instance due to the blender having a 3-prong plug, but there being only a 2-prong outlet.
- `(2) Lid breaks` is not the result of the user breaking it as they grasp or pull at it, but as a result of, for instance, an existing crack in the lid, or the lid having been “cemented” to the carafe by previous dried food. (Ick!)

Exercise for the Reader

- Go through the other product failures in Figure 1 and make sure you understand how these are product failures and not interaction errors.
- What interaction errors could you add to this US?

Revising usage scenarios

You may find, as you continue through the design roadmap, that your design is “drifting” away from what your US says the intervention should do. It's important to keep all the documentation of your design consistent, so you *must* update your US whenever you notice that it no longer represents accurately how your users will use your intervention.

Remember, however, that you *can* mention objects and structures that are not part of the intervention but are in the context of the general situation for which you are designing.

- It's okay to mention the *food* you put in the blender.
- It's okay to refer to the *fork* you do **not** want a child to stick into the toaster.
- It's okay to mention the briefcase you bring with you into the elevator.
- It's okay to refer to the diapers a parent will store in a baby stroller.

Closing notes

There is nothing magical about the format of US diagrams. It is developed to provide a highly efficient, easily constructed means of representing non-linear information. The format is *required* to help graders and assessors focus on content rather than having to “learn” to parse each team's individual format.

Exercise for the reader: Do you see anything wrong with the sample usage scenarios? Discuss them in class.

Exercise for the reader: Explain each of the following design errors in terms of usage scenarios:

- There are no windows on the outer faces of Toronto City Hall because, when they were originally installed, they were sucked out by prevailing winds because the building is shaped like an airfoil.
- The Dana Porter Library at the University of Waterloo is two storeys shorter than it was supposed to be, because of the weight of the books.

Refer to the coffeemaker case study. It is a good example of how easy it can be to design a functional product that is nonetheless unusable; however, a product that isn't functional cannot possibly be usable. That's why we start with *functional* USs, then move on to usability.

Deliverables

A “complete” US includes a diagram, clearly referencing the design concept, Personas, and SUCs for which it was created, and any explanatory notes that the team deems necessary. Notes may be added after the US diagram, using extended point form and indexing notes using the IDs of each step in the US.

However

Usage scenarios are essentially predictions, and so are of limited reliability. While it is possible to, over time and with sufficient data and revision, develop quite accurate scenarios, it can be a very difficult and time consuming task. It is therefore important to:

- **remain flexible:** you want to be able to adapt your scenarios and resulting design intervention to new data as it becomes available throughout a design process;
- **revisit the scenarios often:** it may be the case that just working on a design intervention is enough to bring to light new aspects of scenarios that you had not thought of originally; and
- **remember the users:** if you change a US, you also have to re-verify *all* the user considerations developed from them, or you risk ending up with an unusable product.

To really understand the problem of designing something, we would need to write out a whole series of usage scenarios. We would want to do this in consultation with our clients and with (at least a sample of) the typical expected users, and with the knowledge of the location and general needs for the intervention (one would expect this information to be at least hinted at in the design brief that the client uses to outline the project). Without this, we will be left guessing about what type of intervention is best for a particular situation.

Unfortunately, in academic settings like this one, such work is just not feasible.

See Also

- Hierarchical task analysis

analysis, tool

1) An ancient precursor to the modern smartphone.