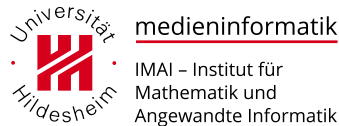


Goal-Directed Design: Scenarios

Setting the Vision

Jörg Cassens

Contextual Design of Interactive Systems



1 Introduction

Assignment 6.1: Go & Carroll

- Required reading for week 3
 - Go, Kentaro, and John M. Carroll. "The blind men and the elephant: Views of scenario-based system design." *interactions* 11, no. 6 (2004): 44-53.
- The texts will be discussed in the tutorial 14.05.2019
- Course readings can be downloaded in the learnweb
- Every text has a wiki-page in the learnweb
 - Use it to describe the text
 - Use it to link the text to the course
- Results of the discussion may also be written up

Where are we?

- Last time: We learned to create personas
- We will now use these personas to generate two design models
 - User stories
 - Scenarios
- User stories:
 - Describe functionality that will be valuable to user of software.
 - A feature request from a user's point of view.
 - It is NOT a bug report or a functional specification.
 - It is NOT a technical design document.
- Scenarios:
 - Stories or scenarios as a means of imagining ideal user interactions
 - Using those scenarios to extract design requirements

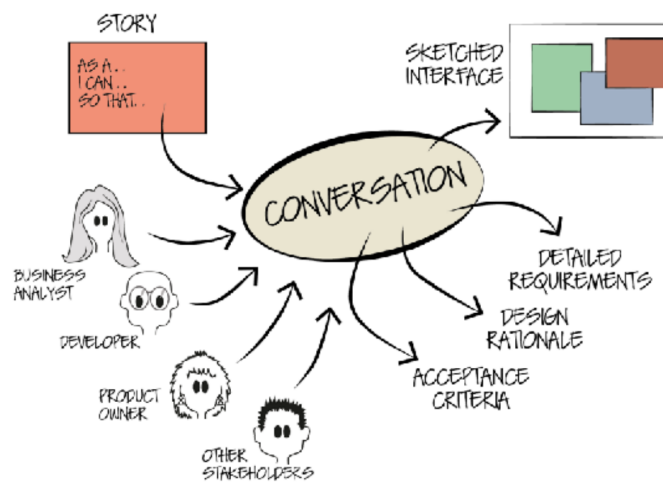
Narrative

- Narrative, or storytelling, one of the oldest human activities
- Power of narrative known to communicate ideas
- However, narrative is also one of our most powerful creative methods
- From a very young age, we are accustomed to using stories to think about possibilities, and this is an effective way to imagine a new and better future for our users
- Imagining a story about a person using our product leverages our creativity to a greater power than when we just imagine a better form factor or configuration of screen elements
- Because of the intrinsically social aspect of narrative, it is an effective and compelling way to share good ideas among team members and stakeholder
- Experiences designed around a narrative tend to be more comprehensible and engaging for users because they are structured around a story

Narration of Interaction

- Narrative also lends itself to effective visual depictions of interactive products
- Interaction design is first and foremost the design of behaviour that occurs over time
- Therefore, a narrative structure combined with the support of fast and flexible visualization tools (such as a whiteboard) is perfectly suited for motivating, envisioning, representing, and validating interaction concepts

Narration as Design Tool



Dell (2018)

2 User Stories

User Stories as a Design Tool

- Shifts the focus from the product to what the eventual owner / user needs or wants it to do.
- A customer buys your product to help them do something.
 - Your stories must help them do it.
- A user has to do something using your product.
 - Your stories must help them do it.
- Remember our focus is on the goal, not the means
 - The narrative should capture what the owner / customer wants to achieve and map that to specific activities

Components

Components

As a [role],

I can [functionality]

so that [rationale]

- Role
 - Persona;
 - important and specific class of user
- Functionality
 - Activity, action, task
- Rationale
 - Reason, motivation
 - The rationale demonstrates the value to the eventual user / owner and determines its priority and the effort to be expended.

Example

Example 1. As a music fan,

I can establish an account,

so that I can legally stream and download music.

- Role is important to the eventual owner
- Rationale demonstrates value to the eventual owner
- Functionality enables the value to be achieved

Scope

- Too broad = impossible to test/code
- Too narrow = more time spent specifying than implementing
- Split long stories (“epics”) into smaller pieces
- Rather than specify small details, get those in conversations with customer & annotate story
- Big stories can serve as placeholders for areas of the system that still need to be discussed

- manage?
- what content?
- control?
- Split epics into smaller stories

Example 2. “As an administrator I can manage posts to the site so that I can control the content our customers see”

2.1 Writing Stories

Writing Stories

- User-centred stories
- In agile approaches, often relegated to customer
 - Written in language of business to allow prioritization
 - Customer is primary product visionary
- Writing good stories needs practice & insight
 - Customer are often not the right stakeholder to write the stories
- Good stories can INVEST

- Independent
- Negotiable
- Valuable to users or customers
- Estimable
- Small
- Testable

Independent & Negotiable

- Independent
 - Each story has it's own value
 - Don't make stories depend on other stories
 - * Stories that depend on other stories are difficult to prioritize and estimate
- Negotiable
 - Stories serve as reminders not contracts
 - Details need to be fleshed out in conversation
 - Stories should have a phrase or sentence to serve as reminder to have conversation & notes about conversation

Valuable & Estimable

- Valuable
 - Both to people using the software and paying for the software
 - Avoid stories valued only by developers (make the benefits to customers/users apparent for these stories)
- Estimable
 - We must have enough information that we can properly size a story so that we plan properly and commit to our work.
 - Reasons why story might not be estimable:
 - * Designers lack domain knowledge
 - * Get details from customer
 - * Story is too big
 - * Split the story into smaller ones

Small & Testable

- Small
 - User Stories should be small enough that they are able to be completed within a sprint.
 - (e.g., A few hours to a few days of design/ development time.)
 - Makes them easy to use in planning
 - Split compound & complex stories
 - Combine too small stories
- Testable
 - All members of the team need a clear and precise way to verify whether or not a User Story has been completed.
 - Can't tell if story is done without tests
 - Tests can be automatable or not

Example

Example 3. as a bank customer i can check my account online so that i can see my balance 24/7

- Why does anyone want to know their current balance?
- This technological solution hides the real requirement

A better story:

as a bank customer i can see my available balance and all anticipated payments for the month so that i can know how much is available to me before next payday

Example

Example 4. as the chicago public health department i want warnings about restaurants that serve raw ingredients so that tourists don't get sick on our dime

- Assumes that ALL restaurants serving raw ingredients will poison their customers
- Only unsafe restaurants are problematic, not all of them
- Why so concerned about tourists? Surely residents are a higher priority.

A better story:

as the chicago public health department i need to know which restaurants serving raw food have failed inspection checks so that we can minimize the possibility of contaminated raw food

Guidelines for Good Stories

- Start with goal stories
- Write closed stories (stories that have a definite end point)
 - “A recruiter can review resumes from applicants to one of her ads” instead of “A recruiter can manage the ads she has placed”
- Size your story appropriately for the time frame it may be implemented in
- Don't rely solely on stories if some things are better expressed in other ways
- Include user roles/personas in stories rather than saying “user”
- Write for a single user (“A Job Seeker” not “Job Seekers”)
- Use Active Voice

Prioritizing Stories

- Assign points to stories based on difficulty / time required
- For each release, customer prioritizes stories
- Developers determine velocity (number of points per release cycle) for previous cycle and plan to implement the highest priority stories up to that number of points for the release

Story responsibilities: Designer

- Focus on story aspects that are
 - Are promises to converse rather than detailed specs
 - Have value to the users or the customer
 - Are independent
 - Are testable
 - Are appropriately sized
- Describing the need for technology / infrastructure in terms of value to users or customers
- Have the conversations with the customers

Story responsibilities: Customer

- Focus on story aspects that are
 - Are promises to converse rather than detailed specs
 - Have value to users or to yourself
 - Are independent
 - Are testable
 - Are appropriately sized
- Have the conversations with the designers/ developers

Example

- Let's fix it

Example 5. as a student i want to check my grades online so that i don't have to wait until the morning

A better story:

as a student i can learn my grades immediately they are available so that i don't suffer longer than necessary

Example

- Let's fix it

Example 6. as a product owner i want to drag and drop all the product backlog items so that i can easily adjust priorities

A better story:

as a product owner i can be made aware of the highest priority needs so that i can align backlog items with priorities

Summary: Why User Stories?

- Emphasize verbal communication
- Comprehensible by everyone
- Right size for planning
- Good for iterative development
- Encourage deferring detail
- Encourage participatory design
- Build up tacit knowledge
- Your story must be right for the solution to be right.
- Think about the non-functional needs – usability, convenience, security, cultural and so on.

Assignment 7.1: User Stories

- Form groups of 4 out of the pairs from Assignment 5.2
- Create a set of 5-10 user stories for your own project.
- Base them on user data you have collected! (not your assumptions)
- Use the general structure
 - As a [role], I can [functionality] so that [rationale]
- For each story you make:
 - Size the story appropriately (Not too big! Not too vague!)
 - Provide info on each role/persona (i.e., not "user")
 - Include non-functional needs (e.g., values, ease-of-use)
- Present your findings in the course

3 Scenarios

Example 7. Suppose an accountant wishes to open a folder displayed on his screen in order to open and read a memo. However, the folder is covered by a budget spreadsheet that he also needs to see while reading the memo. The spreadsheet is so large that it nearly fills the display. The accountant pauses for several seconds, then resizes the spreadsheet, moves it partially out of the display, opens the folder, opens the memo, resizes and repositions the memo, and continues working. (Rosson and Carroll, 2002)

- This is about as routine a work scenario as one could imagine
- Yet even this story conveys important information about window management and application switching:
 - People need to coordinate information sources, to compare, copy, and integrate data from multiple applications; computer displays inevitably get cluttered; and people must find and rearrange windows in these displays.

Scenarios vs. User Stories

- Both methods of describing the user's interaction with a system
- They serve very different functions
 - Scenarios are an iterative means of defining a product's behaviour from the standpoint of specific users (personas)
 - * Does not only include the system's functionality
 - * Priority of functions and how those functions are expressed in terms of what the user sees and how she interacts with the system
 - User stories are exhaustive descriptions of the system's (functional) requirements, focusing on low-level user action and accompanying system response
 - * Use cases permit a complete cataloguing of user tasks for different classes of users
 - * say little or nothing about how these tasks are presented to the user or how they should be prioritized in the interface

Scenarios and Epics

- User stories are used in agile programming methods, but typically they aren't actual stories or narratives
- User stories are much more like informally phrased requirements than they are like scenarios; they don't describe the user's entire flow at a big-picture level or describe what the user's end goal is
- Scenarios are more akin to epics as described by agile methods
- Epics focus more on function and presentation of user interfaces and interactions than they do on user behaviours
- In terms of scope and appropriate level of granularity, they have much more in common with scenarios than user stories do

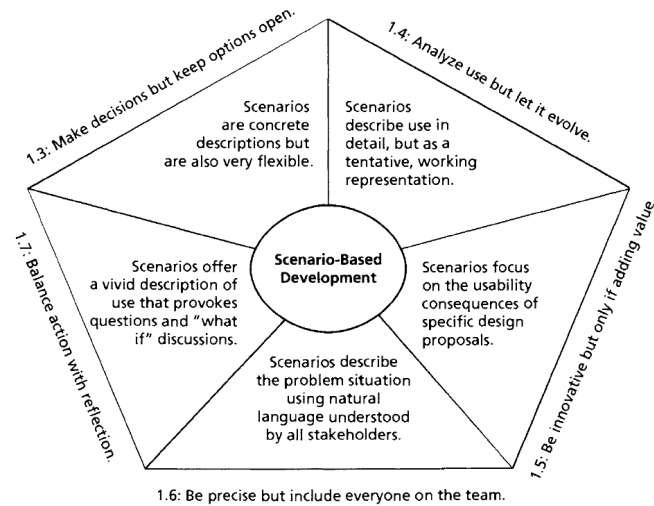
What are Scenarios?

Scenarios

Scenarios are paradoxically concrete but rough, tangible but flexible ... they implicitly encourage "what-if?" thinking among all parties. They permit the articulation of design possibilities without undermining innovation ... Scenarios compel attention to the use that will be made of the design product. They can describe situations at many levels of detail, for many different purposes, helping to coordinate various aspects of the design project. (Carroll, 2000)

- Carroll's use of scenario-based design describes how users accomplish tasks
- It consists of an environmental setting and includes agents or actors who are abstracted stand-ins for users, with role-based names such as Accountant or Programmer

Tradeoffs



(Rosson and Carroll, 2002)

Characteristic Elements

- Rosson and Carroll (2002) suggest that interaction scenarios should have the following components
 - Setting
 - Actors
 - Task goals
 - Plans
 - Evaluation
 - Actions
 - Events

Setting

Situational details that motivate or explain goals, actions, and reactions of the actor(s)

Example 8. Office within an accounting organization; state of work area, tools, etc., at start of narrative

Actors

Human(s) interacting with the computer or other setting elements; personal characteristics relevant to scenario

Example 9. Accountant using a spreadsheet package for the first time

Task goals

Effects on the situation that motivate actions carried out by actors(s)

Example 10. Need to compare budget data with values questioned in memo

Plans

Mental activity directed at converting a goal into a behaviour

Example 11. Opening the memo document will give access to memo information; resizing one window will make room for another

Evaluation

Mental activity directed at interpreting features of the situation

Example 12. A window that is too large can be hiding the window underneath; dark borders indicate a window is active

Actions

Observable behaviour

Example 13. Opening memo document; resizing and repositioning windows

Events

External actions or reactions produced by the computer or other features of the setting; some of these may be hidden to the actor(s) but important to scenario.

Example 14. Window selection feedback; auditory or haptic feedback from keyboard or mouse; updated appearance of windows

Aspects

- Scenarios have a plot; they include sequences of actions and events, things that actors do, things that happen to them, changes in the setting, and so forth
- These actions and events may aid, obstruct, or be irrelevant to goal achievement
- Resizing and moving the spreadsheet supports the goal of opening the folder
- Resizing and moving the memo displays it in a way that enables simultaneous viewing with the spreadsheet
- Pausing does not contribute directly to task goals, although it suggests that the accountant's actions were not completely fluent or automatic

Shortcomings

- [Cooper et al. \(2014\)](#) state that [Rosson and Carroll \(2002\)](#) concept of the actor as an abstracted, role-oriented model is insufficiently concrete to provide understanding of or empathy with users
 - Their scenarios jump too quickly to the elaboration of tasks without considering the user's goals and motivations that drive and filter these tasks
 - Goals are only relevant for specific scenarios, advancing the completion of specific tasks
- The missing ingredient are personas, a tangible representation of the user that acts as a believable agent in the setting of a scenario
- Because personas model goals and not simply tasks, the scope of the problems addressed by scenarios can be broadened to include those related to product definition
 - They help answer the questions "What should this product do?" and "How should this product look and behave?"

Persona-based scenarios

- Persona-based scenarios are concise narrative descriptions of one or more personas using a product or service to achieve specific goals
- They allow us to start our designs from a story describing an ideal experience from the persona's perspective, focusing on people and how they think and behave, rather than on technology or business goals
- Scenarios can capture the non-verbal dialogue between the user and a product, environment, or system over time, as well as the structure and behaviour of interactive functions
- Goals serve as a *filter for tasks* and as a *guide for structuring* the display of information and controls during the iterative process of constructing the scenarios

Role of research

- Scenario content and context are derived from information gathered during the Research phase and analysed during the Modelling phase
- Designers perform a type of role play in creating these scenarios, *walking the personas* through their future interactions with the product or service
- This process leads to real-time synthesis of structure and behaviour and later informs the detailed look and feel
- Finally, personas and scenarios are used to test the validity of design ideas and assumptions throughout the process

Three Types

- Goal-Directed Design employs three types of persona-based scenarios through the design process, each with a successively more interface-specific focus
 - Context scenario
 - Key path scenario
 - Validation scenario
- *Context scenarios* are used to explore, at a high level, how the product can best serve the needs of the personas
- They are created before any design sketching is performed
 - Written from the persona's perspective, focusing on human activities, perceptions, and desires
- Once functional and data elements are defined, a context scenario is revised into a *key path scenario*
 - More specifically describing user interactions
- *Validation scenarios* test the design solution in a variety of situations
 - Tend to be less detailed and typically take the form of a number of what-if questions about the proposed solutions

3.1 Requirements Definition

What?

- The Requirements Definition phase determines the *what* of the design
 - Specifically: what activities, information and interaction capabilities our personas require to accomplish their goals
- Define and agree on the *what* before we move on to the *how* the product looks, behaves, operates, and feels
 - Conflating these two questions can be a pitfall in the design of an interactive product
 - Do not be tempted to jump right into detailed design and render possible solutions

- It runs the risk of leading to a never-ending cycle of iteration
- Proposing a solution without clearly defining and agreeing on the problem leaves you without a clear, objective method of evaluating the design's fitness
 - This in turn can lead to “I like” versus “you like” subjective differences within the product team and stakeholders, with no easy way to converge to consensus

Design Requirements

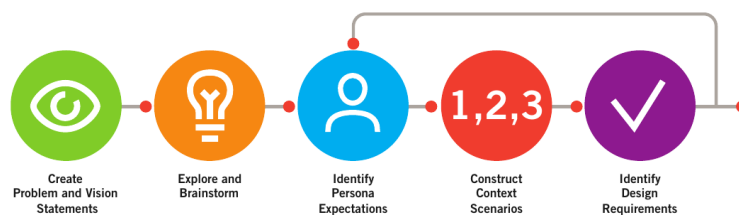
Design Principle

Define what the product will do before you design how the product will do it.

- Design requirements aren't features
 - But there is a relationship between requirements and functions
 - Design requirements are more synonymous with needs
- Design requirements aren't specifications
 - Specifications are focusing on the how, not the why
 - Specifications are of little use for non-developers
- Design requirements are strategic
 - Separating problem and solution provides maximum flexibility in the face of changing technological constraints and rising opportunities
 - By defining user needs, designers can work with technologists to find the best viable and feasible solutions

Requirements Definition Process

- Translation of our models into design solutions consists of two major phases
 - The *Requirements Definition*, described here, answers the broad questions about *what* a product is and *what* it should do
 - The *Framework Definition*, described in the next chapter, answers questions about *how* a product behaves and *how* it is structured to meet user goals



(Cooper et al., 2014)

Requirements Definition: Overview

- The Requirements Definition process consists of the following five steps
 1. Create problem and vision statements
 2. Explore/brainstorm
 3. Identify persona expectations
 4. Construct context scenarios
 5. Identify design requirements
- Although these steps proceed in roughly chronological order, they represent an iterative process
- Expect to cycle through Steps 3 through 5 several times until the requirements are stable

Create problem and vision statements

- Problem and vision statements provide a mandate for the design process
- Helpful in building consensus among stakeholders before the design process moves forward
- *Problem statement* defines the purpose of the design initiative
 - Should reflect a situation that *needs changing*, for both the personas and the business providing the product
 - Often, cause-and-effect relationships exist between business concerns and persona concerns
- *Vision statement* is an inversion of the problem statement that serves as a high-level design objective or mandate
 - It starts with the user's needs, and transitions to how the design vision meets business goals

Example

Problem statement

Company X's customer satisfaction ratings are low. Market share has diminished by 10 percent over the past year because users have inadequate tools to perform tasks X, Y, and Z that would help them meet their goal of G.

Vision statement

The new design of Product X will help users achieve G by allowing them to do X, Y, and Z with greater [accuracy, efficiency, and so on], and without problems A, B, and C that they currently experience. This will dramatically improve Company X's customer satisfaction ratings and lead to increased market share.

Explore and brainstorm

- We have been researching and modelling for some time, hard to avoid having developed some preconceptions about what the solution looks like
- However, we'd like to create context scenarios without these prejudgements, and instead really focus on how our personas would likely want to engage with the product
- We brainstorm at this stage to get these ideas out of our heads so that we can record them and thereby "let them go" for the time being
- Primary purpose: eliminate as much preconception as possible
- A side benefit of brainstorming at this point in the process is that it switches your brain into "solution mode."
- Exploration, as the term suggests, should be unconstrained and uncritical
- Don't spend too much time on the brainstorming step; A few hours for simple projects to a couple of days for a project of significant scope or complexity

Identify persona expectations

- Remember: a person's mental model is her own internal representation of reality—how she thinks about or explains something to herself
- Result of a lifetime of cumulative experiences
- People's expectations about a product and how it works are highly informed by their mental model
- It is important that the represented model of our interfaces—how the design behaves and presents itself—should match what we understand about users' mental models as much as possible
- The represented model should not reflect the implementation model
- To accomplish this, we formally record these expectations

Expectations

- Attitudes, experiences, aspirations, and other social, cultural, environmental, and cognitive factors that influence the persona's expectations
- General expectations and desires the persona may have about the experience of using the product
- Behaviours the persona will expect or want from the product
- How that persona thinks about basic elements or units of data
- Things to look for
 - What do the interview subjects mention first?
 - Which action words (verbs) do they use? What nouns?
 - Which intermediate steps, tasks, or objects in a process don't they mention?
 - * These might not be terribly important to how they think about things.

Construct context scenarios

- All scenarios are stories about people and their activities, but *context scenarios* are the most story-like of the three types in GDD
- A context scenario *tells the story of a particular user persona*, with various motivations, needs, and goals, using the future version of your product in the way that is most typical for that persona
- Describes the broad context in which usage patterns are exhibited
- Context scenarios establish the primary touch points that each primary and secondary persona has with the system (and possibly with other personas) over the course of a day or some other meaningful length of time
- They do not describe product or interaction detail but rather focus on high-level actions from the user's perspective

Context Scenarios

- Context scenarios address questions such as the following:
 - In what setting(s) will the product be used?
 - Will it be used for extended amounts of time?
 - Is the persona frequently interrupted?
 - Do several people use a single workstation or device?
 - With what other products will it be used?
 - What primary activities does the persona need to perform to meet her goals?
 - What is the expected end result of using the product?
 - How much complexity is permissible, based on persona skill and use frequency?
- Context scenarios should *not* represent current product behaviours
- They should represent the new world of Goal-Directed products
- Don't worry yet about exactly how things will get accomplished

Example

- First iteration of a context scenario for a primary persona for a personal digital assistant (PDA) type phone, including both the device and its service
 - Our persona is Vivien Strong, a real-estate agent in Indianapolis, whose goals are to balance work and home life, close the deal, and make each client feel like he or she is her only client
1. While getting ready in the morning, Vivien uses her phone to check her e-mail. Because it has a relatively large screen and quick connection time, it's more convenient than booting up a computer as she rushes to make her daughter, Alice, a sandwich for school.
 2. Vivien sees an e-mail from her newest client, Frank, who wants to look at a house this afternoon. The device has his contact info, so she can call him with a simple action right from the e-mail.
 3. While on the phone with Frank, Vivien switches to speakerphone so she can view the screen while talking. She looks at her appointments to see when she's free. When she creates a new appointment, the phone automatically makes it an appointment with Frank, because it knows with whom she is talking. She quickly enters the address of the property into the appointment as she finishes her conversation.
 4. After sending Alice to school, Vivien heads into the real-estate office to gather some papers for another appointment. Her phone has already updated her Outlook appointments, so the rest of the office knows where she'll be in the afternoon.
 5. The day goes by quickly, and eventually Vivien is running a bit late. As she heads toward the property she'll be showing Frank, the phone alerts her that her appointment is in 15 minutes. When she flips open the phone, she sees not only the appointment, but also a list of all documents related to Frank, including e-mails, memos, phone messages, and call logs to Frank's number. Vivien initiates a call, and the phone automatically connects to Frank because it knows her appointment with him is soon. She lets him know she'll be there in 20 minutes.
 6. Vivien knows the address of the property but is unsure exactly where it is. She pulls over and taps the address she put into the appointment. The phone downloads directions along with a thumbnail map showing her location relative to the destination.
 7. Vivien gets to the property on time and starts showing it to Frank. She hears the phone ring from her purse. Normally while she is in an appointment, the phone automatically goes to voicemail, but Alice has a code she can press to get through. The phone knows it's Alice calling, so it uses a distinctive ringtone.
 8. Vivien takes the call. Alice missed the bus and needs to be picked up. Vivien calls her husband to see if he can do it. She gets his voicemail; he must be out of service range. She tells him she's with a client and asks if he can get Alice. Five minutes later the phone sounds a brief tone. Vivien recognizes it as her husband's; she sees he's sent her an instant message: "I'll get Alice; good luck on the deal!"

Magic

- Example scenario remains at a fairly high level, without getting too specific about interfaces or technologies
- Important to create scenarios that are within the realm of technical possibility, but at this stage the details of reality are unimportant
- The activities in the scenario tie back to Vivien's goals and try to eliminate as many tasks as possible
- A powerful tool in the early stages of developing scenarios is to pretend the interface is magic
- If your persona has goals and the product has magic powers to meet them, how simple could the interaction be?
- Products that meet goals with a minimum of hassle and intrusion seem almost magical to users
- It's the goal-directed behaviour, not technology alone, that provides magic

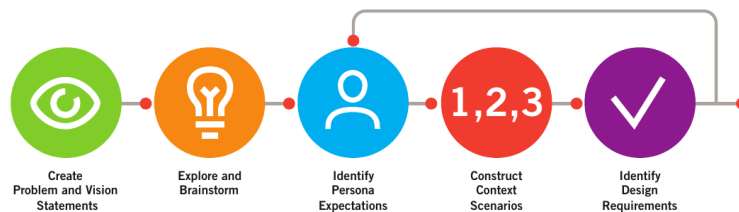
Identify design requirements

- After you are satisfied with an initial draft of your context scenario, you can analyse it to extract the personas' needs or design requirements
- Design requirements can be thought of as consisting of objects, actions, and contexts
- Requirements from the preceding scenario might read as follows:
 - Call (action) a person (object) directly from an appointment (context).

Requirement facets

- **Data**
 - Personas' data needs are the objects and information that must be represented
 - Useful to think of data requirements as the objects and related adjectives
 - * accounts, people, addresses, documents, messages, songs, and images, as well as attributes of those, such as status, dates, size, creator, and subject
- **Functional**
 - Operations or actions that need to be performed on the system's objects; typically translated into interface controls
 - Can be thought of as the product's actions
- **Contextual**
 - Relationships or dependencies between sets of objects in the system
 - Can include which objects in the system need to be displayed together to make sense for work flow or to meet specific persona goals
 - May include the physical environment the product will be used in (an office, on the go, in harsh conditions) and the skills and capabilities of the personas using it

Requirements Definition Process



(Cooper et al., 2014)

4 Scenario-Based Design

- A very rough sketch could be used to prototype an activity scenario
- Details of system interaction would not be specified, but people could evaluate the sketch in the context of the scenario, critique it, act out the scenario, explain it to peers, and so on
- SBD distinguishes between formative evaluation, which is carried out to guide redesign, and summative evaluation, which serves a system verification function

SBD & GDD Scenarios

- The different kinds of scenarios in both design processes do not have direct counterparts
- However, as a first approximation, we can establish the following relations
 - SBD: Problem Scenario
 - * There is no counterpart in GDD since the scenarios explicitly focus on the new system
 - * Problem scenarios can be included in GDD as a precursor to context scenarios
 - SBD: Activity Scenario
 - * The closest approximation would be the GDD *context scenario*
 - SBD: Information and Interaction Scenarios
 - * In GDD, the successive development of *key path scenarios* are the closest equivalent
 - SBD: Usability Specifications
 - * Related to the GDD *validation scenarios*, but the usability specifications are more targetted at end user tests

Claims Analysis

- In SBD, the analysis and refinement of scenarios is stimulated by claims, statements that list important features of a situation and their impacts on users' experiences
- In requirements analysis, these features are elements in the current situation; as the scenario content shifts from analysis to design, the claims call out features of the proposed solution
- Claims are related to the general notion of tradeoffs in design, because they always analyse both positive and negative usability impacts
- The analysis of claims organizes and documents the "what- if" discussions the design team carries out when considering and prioritizing alternatives

Claims

- Claims elaborate a set of scenarios, explaining how and why a particular feature is having a range of impacts on the personas
- A claims analysis documents why one or more scenarios were written, by isolating the most important features of the narratives
- The claims extend the scenarios, pointing to possible effects a feature might have in other scenarios (i.e., without writing out a new scenario)
- Claims analysis promotes a balanced view of a situation
- Each feature is analyzed to consider both positive impacts (prefaced with plus signs) and negative impacts (prefaced with minus signs)
- The claims motivate design reasoning—designers will try to increase positive impacts while decreasing negative impacts

Example

Exhibits composed of physical elements	+ leverages existing skills at grasping, moving, and constructing things in the world + encourages direct engagement and involvement by visitors – each element consumes a fixed amount of physical space – some pieces may be expensive or difficult to produce or copy
Repeated involvement by the same students	+ increases student familiarity and competence from year to year + encourages formation of a “science fair community” – but inexperienced students may find it hard to “break in”
Holding the fair in the high school gym	+ reinforces the feeling of community and school interaction + may bring back memories of other school events held in this place – but the organizers may need to coordinate with other school events (e.g., sports) – but the gym may not be set up with fair-specific resources (e.g., electricity)

5 Tutorial

Assignment 5.2: Contextual Inquiry

- Form pairs of 2 who do not know each other well
- Use a system you are acquainted with and research your use of it
 - You actually have to do the task
 - You would actually do the task on campus on the device you’re using
- The other person conducts a contextual inquiry on their task:
 - Focus. Decide what to pay attention to.
 - Partnership. You act as an interested learner, they act like a knowledgeable expert.
 - Perform the inquiry. Ask probing questions. Have them teach you. Don’t generate questions in advance; think of them as you observe. Focus questions on what you see happening in context.
- Document your findings
- Present your findings in the course
 - Due date: 07.05.2019

Assignment 6.2: Persona Construction

- Form groups of 4 out of the pairs from Assignment 5.2
- Imagine you are re-designing the system used in the previous assignment
- Create at least two different personas
 - For example a primary and a negative
- Focus on
 - Characteristics
 - Experiences
 - Motivations
 - Goals
- Feel free to use a templates
- Present your findings in the course
 - Due date: 21.05.2019

Assignment 7.2: Scenario

- Form groups of 4 out of the pairs from Assignment 6.2
- Create a context scenario for each of your personas
- Make sure you cover the typical aspects
 - Setting
 - Actors (Personas)
 - Task goals
 - Plans
 - Evaluation
 - Actions
 - Events
- Present your findings in the course

References

Literatur

Carroll, J. M. (2000). *Making use: scenario-based design of human-computer interactions*. MIT press.

Cooper, A., Reimann, R., Cronin, D., and Noessel, C. (2014). *About Face (fourth edition): the essentials of interaction design*. John Wiley & Sons.

Dell, N. (2018). Hci and design. Course material, Cornell University.

Go, K. and Carroll, J. M. (2004). The blind men and the elephant: Views of scenario-based system design. *interactions*, 11(6):44–53.

Rosson, M. B. and Carroll, J. M. (2002). *Usability Engineering: Scenario-Based Development of Human-Computer Interaction*. Morgan Kaufmann/Academic Press.