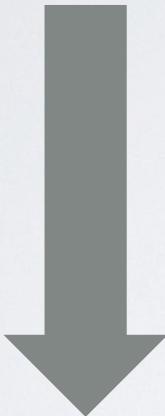


# **User Interface Design & Evaluation**

---

Scenarios & Scenario-Based Design

# Overview



- Personas
- Scenarios
- Scenario-based design

# Scenario-Based Design?

- Development Carroll and colleagues (see Rosson and Carroll, 2002)
- Evolved from the difficulties of using multidisciplinary teams for system development

His example is developing a virtual school system - you need teachers, curriculum experts (e.g. people who write textbooks), cognitive and educational psychologists, software engineers, networking experts etc !

- How can all these people work together and understand each other's visions and problems?!

# Overview

## Scenarios

- Problem scenarios  
(describes current practice)
- Activity scenarios  
(Describes ideas of how to meet the users needs)

## Use Cases

# What are Scenarios?

- Short stories of use (about the personas)
- Focus on named users of the system
- Rich in detail of context and feelings
  - Where, when, who else is around
  - User experience and emotion
- Explains goals to be achieved

# What are Scenarios?

A design technique used to envision future use of a system

Focusing on how users can achieve their goals

Helps designers & developers understand how system will really be used

A story about a particular persona interacting with the system

May be based on a use case, or a set of use cases

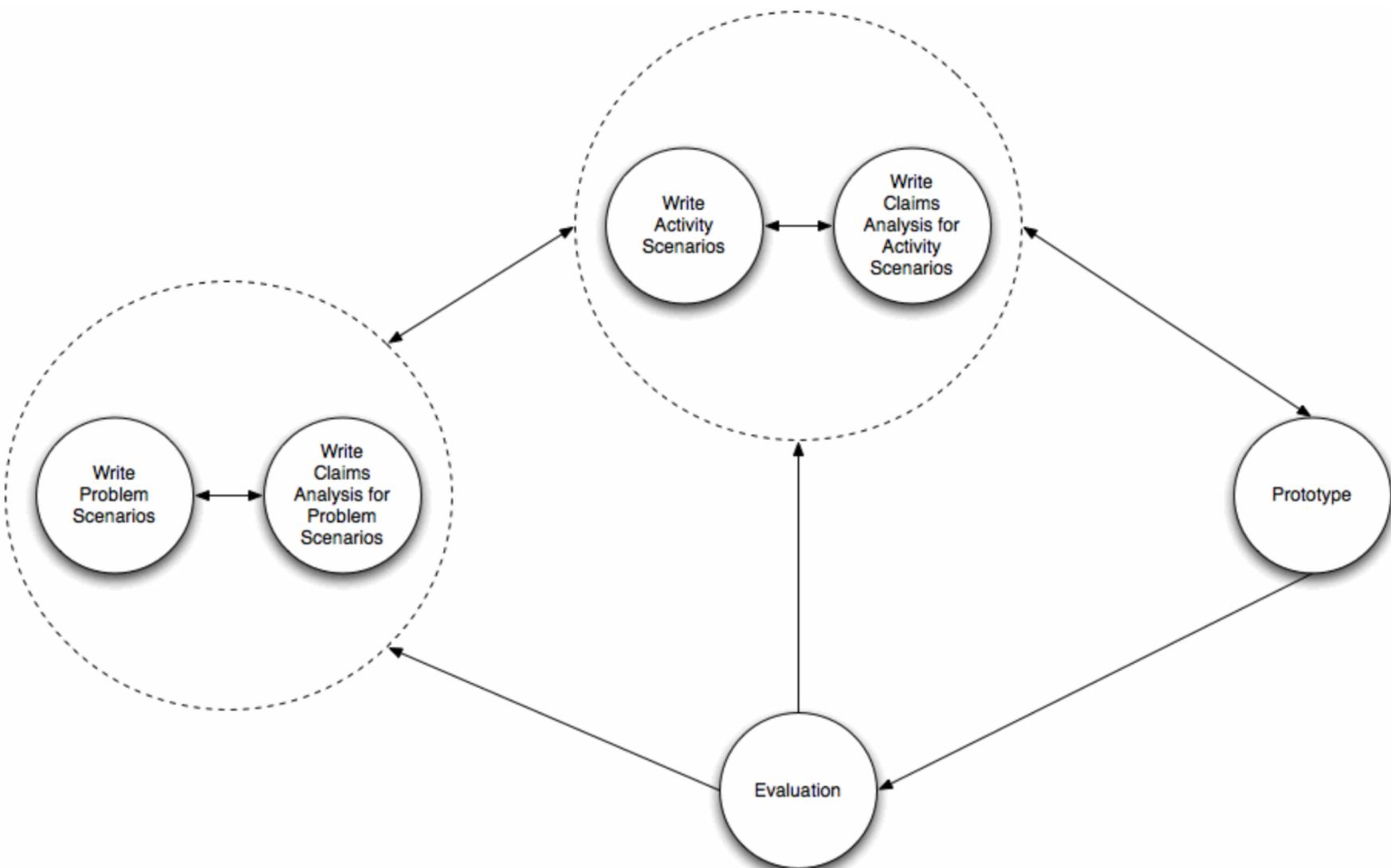
Can be used for usability testing

Scenarios become progressively more detailed

# What are Scenarios?

- Elaborate the design by writing scenarios (=stories) about the current domain and existing interactions!
- These stories can be analyzed for the benefits and detractors for the user in accomplishing their goals!
- These start by being quite simple, but increase in complexity as the design is further articulated

# Where do they fit ?



# Key Elements of Scenarios

Element	Definition	Example
Setting	Situation elements that motivate or explain goals, actions, and reactions to the actor(s)	Office within an accounting organization, state of work area, tools etc at the beginning of the scenario
Actors	Humans interacting with the technology or other setting elements, personal characteristics relevant to the scenario	Accountant using a spreadsheet package
Task goals	Effects on the situation that motivate actions carried out by the actor (s)	Need to compare budget data with values questioned in a memo

# Key Elements of Scenarios

Element	Definition	Example
Plans	Mental activity directed at converting a goal into a behaviour	Opening the memo document will give access to memo information, re-sizing windows to allow viewing of memo and spreadsheet
(User) Evaluation	Mental activity directed at interpreting features of the situation	A window that is too large can be hiding the window underneath; dark borders indicate a window is active

# Key Elements of Scenarios

Element	Definition	Example
Actions	Observable user behaviour	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 the scenario	Window selection feedback; auditory or haptic feedback from keyboard or mouse; updated appearance of window (perhaps unnoticed by user)

# EXAMPLE ATM SCENARIO

- It's Friday afternoon and Joe is flying to Sydney. He doesn't have enough money for a taxi to the airport, and he's running late. He goes to the local ATM and identifies himself. He specifies that he wants \$100 from his savings account. He'd like the money in \$20 notes so that he can give the taxi driver the correct change. He doesn't want a printed receipt, as he doesn't bother keeping track of transactions in this account.

# VALUE OF SCENARIOS

- Communication with stakeholders
  - Natural way for users to talk
  - Uses appropriate language
- Emphasises user goals and user experience
- Useful throughout the design process

# ELEMENTS OF SCENARIOS

- Setting
  - Situation, context
- Actors
  - Characters who use the system
- Task goals/objectives
  - Intentions, motivations
- Plans
  - Mental intention of actions
- Evaluation
  - Mental interpretations by actors
- Actions
  - Observable behaviour
- Events
  - External actions of the system, context

# PROBLEM SCENARIOS

- Outline a situation in which the system is not used but would be useful
- User experience of current practice naturally produces problem scenarios
- Illustrates constraints, irritations or issues with existing techniques
- Describes the problem domain, not necessarily a problem to be solved

# EXAMPLE PROBLEM SCENARIO

- Jane is required to post her master's project proposal and she is looking for ideas. She surfed the Web and was hoping to find some inspiration. But she found herself spending hours doing research. She was not very productive since the information she was looking for was scattered on many different Web sites. She thinks it'll be really useful to her if there is one site that will provide information on all of the design.

# ACTIVITY SCENARIOS

- Explain an imagined usage of the new system
- Concentrate on high level issues, not how it actually works
- Focus designs on what is to be achieved with the system without prematurely committing to a design

# EXAMPLE ACTIVITY SCENARIO

- Jackie wants to fly to Århus next Thursday, returning on the last flight on Friday. She wants to know how much this would cost, and whether it would be cheaper to take a different flight back. She is not quite sure how Århus is spelt on the computer. When she has found the right flight, she wants to confirm the purchase with a credit card.

# MORE DETAILED ACTIVITY SCENARIO

Too much technical detail?

- Luisa Martinez stretches, trying to wake up her muscles. It's cold, dark, and it's too damn early to be doing this. Grabbing her run pack—her specs, a Nalgene bottle of water—she heads out into the San Francisco morning.
- On the sidewalk, she turns on the Web tablet and puts on the specs. They're new; she got them as a gift from her brother, who loves the latest gadgets. They're supposed to work with the new networks that the City is installing (yet another Google project, she recalls), as well as provide a new way of controlling her tablet (a pocket-size device replacing an old iPod she recycled a month or three ago).
- She sees a slowly opening flower off to the side of her vision as the system comes online, a rose. She smiles; Diego must have done some customization before he sent it. A text crawl, just below her eye-line, asks her to confirm if she wants to use her "Jogging" profile from the tablet. She pauses—was this a voice interface?—and remembers that the specs have one of those accelerometers in it to respond to movement. Hesitantly, she nods, the display clears, and she sets off.

# Worked Example: Physics Class

*Matilda, Dexter, Hildegarde, and Godfrey carry out a physics lab on inelastic collisions, in which two miniature cars collide and the resulting car velocities are measured using photo gates.!*

*The students collaborate to set up the experiment: Hildegarde collects and positions the cars, Dexter and Godfrey construct the apparatus.!*

*The group executes the experiment collaboratively: Dexter collides the two cars together, Matilda reports the photo gate results, and Godfrey catches the cars after they collide.!*

*As data accrues, the students discuss how to interpret their findings. They collectively perform calculations and document their results on worksheets provided by their teacher!*

# **Claims Analysis: Feature and its advantages and disadvantages**

## **Feature: Physical inelastic collision experiment !**

- Pros: practices collaborative skills; practices negotiation of tasks and roles; provides concrete, hands-on experience; resembles real-world situations !
- Cons: but experiment apparatus may not be available; but conditions may be hard to test; but students at different sites cannot work together

# **Claims Analysis: Feature and its advantages and disadvantages**

## **Feature: Joint execution of experiment !**

- Pros: practices collaborative skills; allows students to take on more complex and rewarding experiments; involves students in peer learning

# **Claims Analysis: Feature and its advantages and disadvantages**

## **Feature:** Joint interpretation and analysis of data !

- Pros: practices collaborative skills; may improve analysis results by pooling ideas; involves students in peer learning !
- Cons: but students at different sites can not work together!

# **Activity Scenario**

*Mertyl, Ethel, and Sandy - middle school students in Ms. Snodgrass' science class -have elected to work together on a force and motion assignment in which the group is to execute a computer-based block and plane simulation and graph the simulation results.*

*Terri-a high school physics student-has been assigned to mentor them for this assignment.*

*Each group completes the assignment over a two-day period, with mentoring occurring in the thirty minutes of overlap time between the middle and high school classes. Different groups of middle school students rotate through three computers over several days.*

## **Activity Scenario 2**

*Mertyl, Ethel, and Sandy meet Terri over the computer, using a video conferencing package.-Terri guides the middle school students through the block-and-plane simulation.*

*She steps through the simulation on her computer, describing the process as she proceeds.*

*She occasionally picks up the camera and aims it towards her screen to show the middle schoolers the state of the simulation on her screen.*

*Sometimes Ten-i takes a snapshot of her screen and copies it onto an electronic whiteboard, annotating or highlighting specific features of interest.*

# Claims Analysis for Activity Scenarios

- For Activity Scenarios, we are looking for the same kinds of features of the scenario
  - however this time we are paying close attention to suggested solutions regarding how they have improved or worsened the previous situation!

# Claims Analysis: Features in Activity Scenario

## Feature: Block and plane simulation !

- Pros: allows multiple, concurrent executions; allows testing of boundary conditions; supports rapid data collection !
- Cons: but concurrent execution may be distracting; but mapping to real-world phenomenon may not be obvious

# Claims Analysis: Features in Activity Scenario

## Feature: Electronic whiteboard !

- Pros: offers a salient space for collaboration; supports sharing of text and images; may increase awareness of local collaborators' activities (via telepointers) !
- Cons: but students may erase or overwrite each other's work, but may support limited media or imports from other useful applications; but may distract students from other activity!

# Claims Analysis: Features in Activity Scenario

## Feature: Video conferencing !

- Pros: increases remote presence; makes learning experience more vivid, engaging; is natural to students !
- Cons: but may provide poor sound; but students may be especially frustrated when technical problems occur

# Claims Analysis: Features in Activity Scenario

**Feature:** Small number of computers in classroom!

- Pros: makes the technology less physically intrusive !
- Cons: requires students to time-share computers; prohibits class-wide activities

# Claims Analysis: Features in Activity Scenario

## **Feature:** Short, overlapping class periods

- Pros: reflects typical scheduling of classes
- Cons: but restricts synchronous collaboration with remote sites!

# Iterate through the lifecycle

- Either iterate around and write more scenarios exploring advantages and disadvantages

or

- Interaction design and prototyping carrying what has been learned from the scenarios forward into the design

# Scenario Based Design Advantages

## Advantages:

- Allows quite detailed development of the design without commitment to coding or even prototyping!
- Easy to change requirements, consider alternative designs!

# Scenario Based Design Disadvantages

Does not give users hands-on experience with even low fidelity prototypes which may inform their views!

# User Interface Design

“Design is the conscious and intuitive effort to impose meaningful order”

Interaction design AND visual design

How does it behave?

What does it look like?

How does it make users feel?

Wireframes, mock-ups, and/or prototypes

# Defining the UI framework

## High level design

What pages do we have?

What panes need to exist within the pages and how do they work together?

What design elements are included in each page, pane, etc.?

Should be a holistic view of the design, not too detailed

## Key path scenarios

Allows for iterating on the details

Start talking about technical feasibility

## Context Scenario

- Lisa is in lecture and realizes she's confused when the instructor starts talking about mitosis. She takes note of the time.
- Later that day she opens up her uni course site and goes directly to the webcast for that day and reviews the portions of lecture via the webcast she needed clarification on.

## Key Path Scenario

- Lisa is in lecture and realizes she's confused when the instructor starts talking about mitosis. She takes note of the time.
- Later that day she opens up her uni course site **clicks on the “Most Recent Webcast” link. Uni course switches to the “Use Webcast” View and the webcast for the day plays.**
- **Lisa looks at her notes to see the time she noted earlier, and enters it into the “Lecture Time” field and presses “Enter.” The lecture jumps forward to the point where the instructor was talking about mitosis.**

# Development

- Constant communication  
No throwing it over the wall
- Continuous iterations as we learn more from development

# Practical

# DEVISE AN ACTIVITY SCENARIO

- Consider again, the proposed University news app.
- Write a short activity scenario describing one possible use of it.

# USE CASES

- Narrower scope than a scenario
  - Focus just on the interaction of a user with a system
- More technical
  - No context
  - No user experience
- Should still work at a high level
  - “The user chooses a country”
  - Not “The user selects a country from a drop down menu with the mouse”

# USE CASES

Narrative of achieving a goal

Variety of forms

Actor, system, goal

# USE CASES

System is black box

Articulate a goal

Narrative

But...

**...use cases are not scenarios**

User is also a black box  
Real goals? Real behaviour?

No context  
eg security, external events

Specify interaction/objects

# Essential use cases

Equality of system and user

Higher abstraction

Intentions not actions

# Example: send email

User intention	System responsibility
1. Write email	
	2. Enable text entry
3. Specify addressees	
	4. Offer addresses
5. Send	
	6. Ensure delivery

# Practical

Specify an essential use case for a person looking through the car park summaries for the departments and working out where Fine Arts is.

# User-centred?

Not system centred  
Nonspecific user  
No context

# EXAMPLE USE CASE

1. The system provides a new editor area/box filled with all the article's relevant content with an informative edit summary for the user to edit. If the user just wants to edit a section of the article, only the original content of the section is shown, with the section title automatically filled out in the edit summary.
2. The user modifies the article's content till satisfied.
3. The user fills out the edit summary, tells the system if he/she wants to watch this article, and submits the edit.
4. The system saves the article, logs the edit event and finishes any necessary post processing.
5. The system presents the updated view of the article to the user.

# ALTERNATIVES IN A USE CASE

- Normal course
  - The most common sequence of actions
- Alternative course
  - Other alternatives that might happen
  - Frequently used for error handling

# EXAMPLE ALTERNATIVE COURSE

- :
- 2. The user modifies the article's content till satisfied.
- :
- 5. The system presents the updated view of the article to the user.

Alternatives:

- 2. The user selects Cancel.
  - 2.1 The system discards any change the user has made, then goes to step 5.

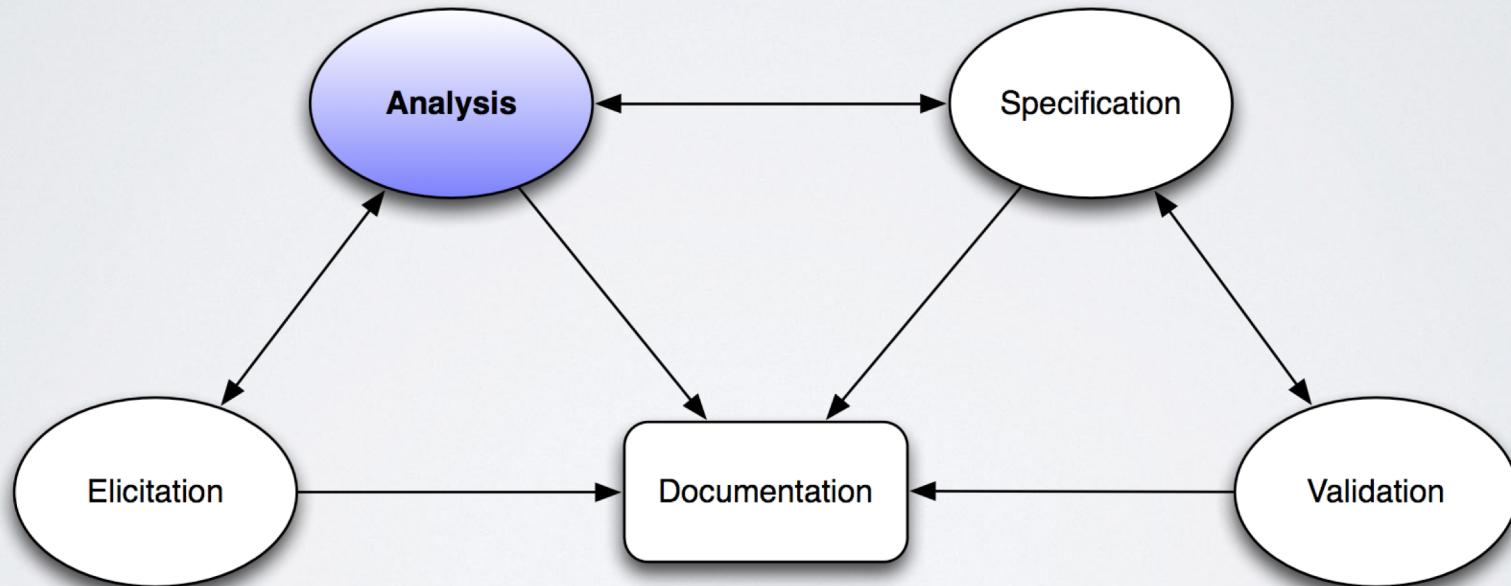
# CREATING A USE CASE

- Take the scenario you devised earlier and turn the **user-system interaction** part of it into a use case.
- Don't forget to consider possible alternative cases.
- (If your scenario is too complex or trivial, do a use case for another task.)

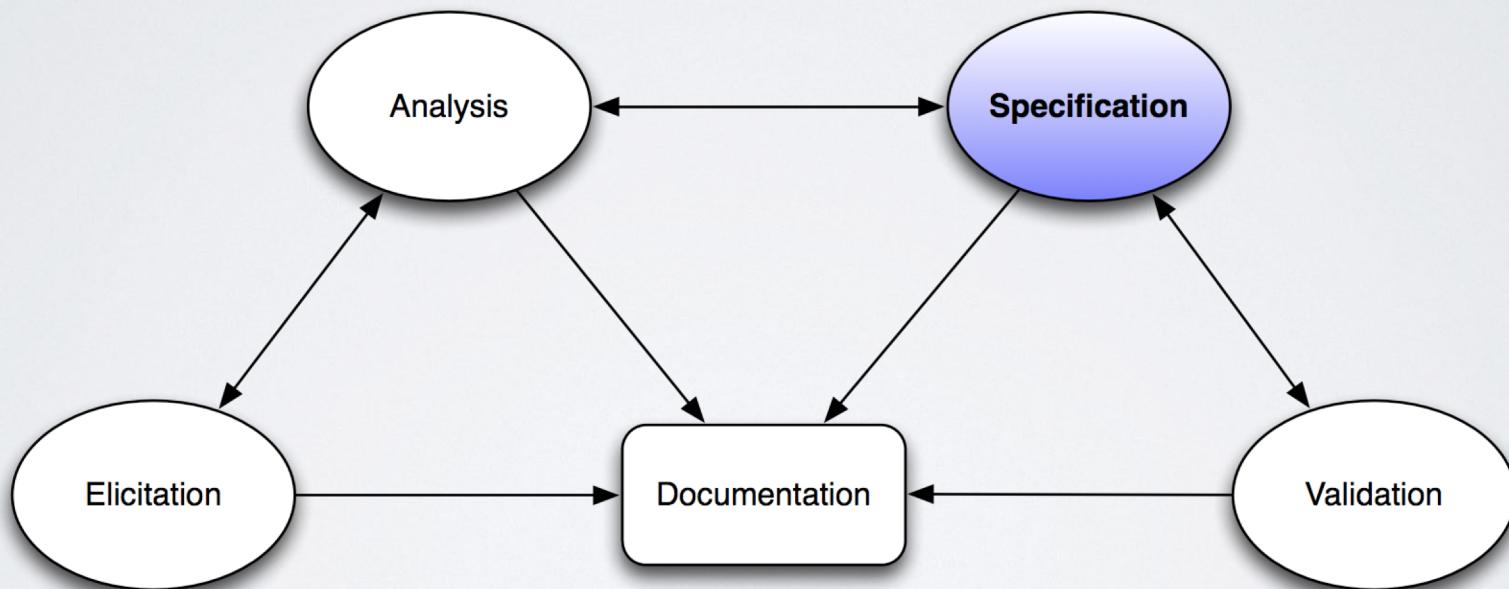
# REQUIREMENTS ANALYSIS: USE CASES TO REQUIREMENTS AND BACK AGAIN

Congo the Online Shop

# WHERE ARE WE IN THE MODULE?



# WHERE ARE WE GOING IN THE MODULE?



# ANALYSIS-SPECIFICATION- ANALYSIS

- The above diagrams are a bit of a lie ... requirements very seldom enter one state and go into another and never go back again.
- Each of the analysis tools that you learn in this course are used to specify new requirements or to **refine existing requirements** (the latter is more often the case in reality).

# CASE STUDY: ONLINE SHOPPING

- In order to demonstrate how this might work on a (semi) complex application, consider an online book store *Congo*.
- Congo follows the tried and true business model of having customers add products to a shopping cart, checking it out, paying, and shipping to an address.

# ONLINE SHOPPING: INITIAL REQUIREMENTS

1. The user will be able to add products to the shopping cart.
2. The user will be able to view the contents of the shopping cart.
3. The user will be able to pay for the contents of their shopping cart.
4. The user will be able to choose an address to ship the products.
5. Congo's online inventory will be updated.
6. The warehouse will ship the order within 3 days of the order.

# HOW DO REFINE THESE REQUIREMENTS?

- Clearly this set of requirements is incomplete.
- They may have come from original elicitation sessions, been inherited or ~~stolen~~ borrowed from other competitors.
- They may have come from goal-based analysis.
- We need to analyse these for gaps, identify different actors, and determine what else the system must do.

# USE CASES: IDENTIFY NOUNS

1. The **user** will be able to add **products** to the **shopping cart**.
2. The **user** will be able to view the contents of the **shopping cart**.
3. The **user** will be able to pay for the contents of their **shopping cart**.
4. The **user** will be able to choose an **address** to ship the products.
5. **Congo's** online **inventory** will be updated.
6. The **warehouse** will ship the order within 3 days of the order.

- Some of these nouns are actors.
- Those nouns that perform actions in the system are actors.
- How do we identify actors?
- Look for verbs

# USE CASES: IDENTIFY ACTORS

- Actors in this system include:
  - User – Too generic we can rename them to be a “Customer” based on their functionality
  - Warehouse – Is it the warehouse itself that is doing the shipping? Is there a shipping company we should worry about? Do we need more information?
  - Congo – Too broad; looking at what Congo does in the initial description, they are having just their inventory updated; let’s designate a system actor to look after that: the Congo Inventory System (CIS)

# USE CASES: REFINE REQUIREMENTS

- 1. The customer will be able to add products to the shopping cart.
  - 2. The customer will be able to view the contents of the shopping cart.
  - 3. The customer will be able to pay for the contents of their shopping cart.
  - 4. The customer will be able to choose an address to ship the products.
  - 5. The CIS inventory will be updated.
  - 6. The Warehouse will ship the order within 3 days of the order.
- We have already refined the requirements a little bit.
  - We now have a better list of actors for our system.
  - Now what?
  - Identify actions that people will take.

# USE CASES: IDENTIFY ACTIONS

1. The customer will be able to **add products** to the shopping cart.
  2. The customer will be able to **view the contents** of the shopping cart.
  3. The customer will be able to **pay for the contents** of their shopping cart.
  4. The customer will be able to **choose an address** to ship the products.
  5. The CIS inventory will be **updated**.
  6. The Warehouse will **ship the order** within 3 days of the order.
- Each of the actors do different things in the system.
  - We can begin by deciding what broad use cases could represent this.

# USE CASES: IDENTIFY ACTIONS

Requirement	Use Cases
The customer will be able to add products to the shopping cart.	<i>Add product to cart (Customer)</i>
The customer will be able to view the contents of the shopping cart	<i>View cart contents (Customer)</i>
The customer will be able to pay for the contents of their shopping cart.	<i>Pay for cart (Customer)</i>
The customer will be able to choose an address to ship the products.	<i>Choose address (Customer)</i>
The CIS inventory will be updated.	<i>Update inventory (CIS)</i>
The Warehouse will ship the order within 3 days of the order.	<i>Ship order (Warehouse)</i>

# USE CASE ANALYSIS

We can begin writing use cases;

however, some basic analysis can already be completed.

We begin asking questions about the use cases that we have in comparison to information we have collected from our stakeholders.

# CONGO: ACTOR ANALYSIS DO WE HAVE THEM ALL?

There is no existing shopping cart system, so we must develop it; we will need more detail in our requirements for this, and we add a couple of initial use cases *Update contents* and *Empty*.

There is an existing Warehouse Inventory system that will be used; therefore we can assume the Warehouse shipping an item to be sufficient detail. We should record this in our requirements somewhere.

There is an existing CIS, so we can assume “Updating inventory” is sufficient detail, we do not need to explain how the inventory will be updated. Similarly for the payment system.

# USE CASES: UPDATE ACTORS

Requirement	Use Cases
The customer will be able to add products to the shopping cart.	<i>Add product to cart (Customer)</i> <b>Update Contents (Cart)</b>
The customer will be able to view the contents of the shopping cart	<i>View cart contents (Customer)</i>
The customer will be able to pay for the contents of their shopping cart.	<i>Pay for cart (Customer)</i> <b>Empty (Cart)</b>
The customer will be able to choose an address to ship the products.	<i>Choose address (Customer)</i>
The CIS inventory will be updated.	<i>Update inventory (CIS)</i>
The Warehouse will ship the order within 3 days of the order.	<i>Ship order (Warehouse)</i>

# **Analysis: Examine Pre-conditions / Triggers**

Are there any assumptions in your use cases regarding events that have occurred previously?

Example:

The user is allowed to choose an address.

**This implies that there is more than one address.**

We are missing a use case where the user can add addresses.

# Use cases: Update for pre-conditions / triggers

Requirement	Use Cases
The customer will be able to add products to the shopping cart.	<i>Add product to cart (Customer)</i> <i>Update Contents (Cart)</i>
The customer will be able to view the contents of the shopping cart	<i>View cart contents (Customer)</i>
The customer will be able to pay for the contents of their shopping cart.	<i>Pay for cart (Customer)</i> <i>Empty (Cart)</i>
The customer will be able to choose an address to ship the products.	<i>Choose address (Customer)</i> <b>Enter Address (Customer)</b>
The CIS inventory will be updated.	<i>Update inventory (CIS)</i>
The Warehouse will ship the order within 3 days of the order.	<i>Ship order (Warehouse)</i>

# **Analysis: Examine Pre-conditions / Triggers**

Some pre-conditions may not even relate to the actor being examined:

Example:

The Warehouse is to ship things within a certain amount of time; how does it know what to ship? How does it know when the period of time begins? It is triggered by receiving an order.  
**The CIS must send the order through to the Warehouse**

# Use cases: Update for pre-conditions / triggers

Requirement	Use Cases
The customer will be able to add products to the shopping cart.	<i>Add product to cart (Customer)</i> <i>Update Contents (Cart)</i>
The customer will be able to view the contents of the shopping cart	<i>View cart contents (Customer)</i>
The customer will be able to pay for the contents of their shopping cart.	<i>Pay for cart (Customer)</i> <i>Empty (Cart)</i>
The customer will be able to choose an address to ship the products.	<i>Choose address (Customer)</i> <i>Enter Address (Customer)</i>
The CIS inventory will be updated.	<i>Update inventory (CIS)</i>
The Warehouse will ship the order within 3 days of the order.	<b><i>Send order (CIS)</i></b> <i>Ship order (Warehouse)</i>

# ANALYSIS: EXAMINE COMPLEXITY

Use cases are intended to represent atomic actions in the system.

Are any of the use cases doing too much?

Example:

The customer pays for items in the cart.

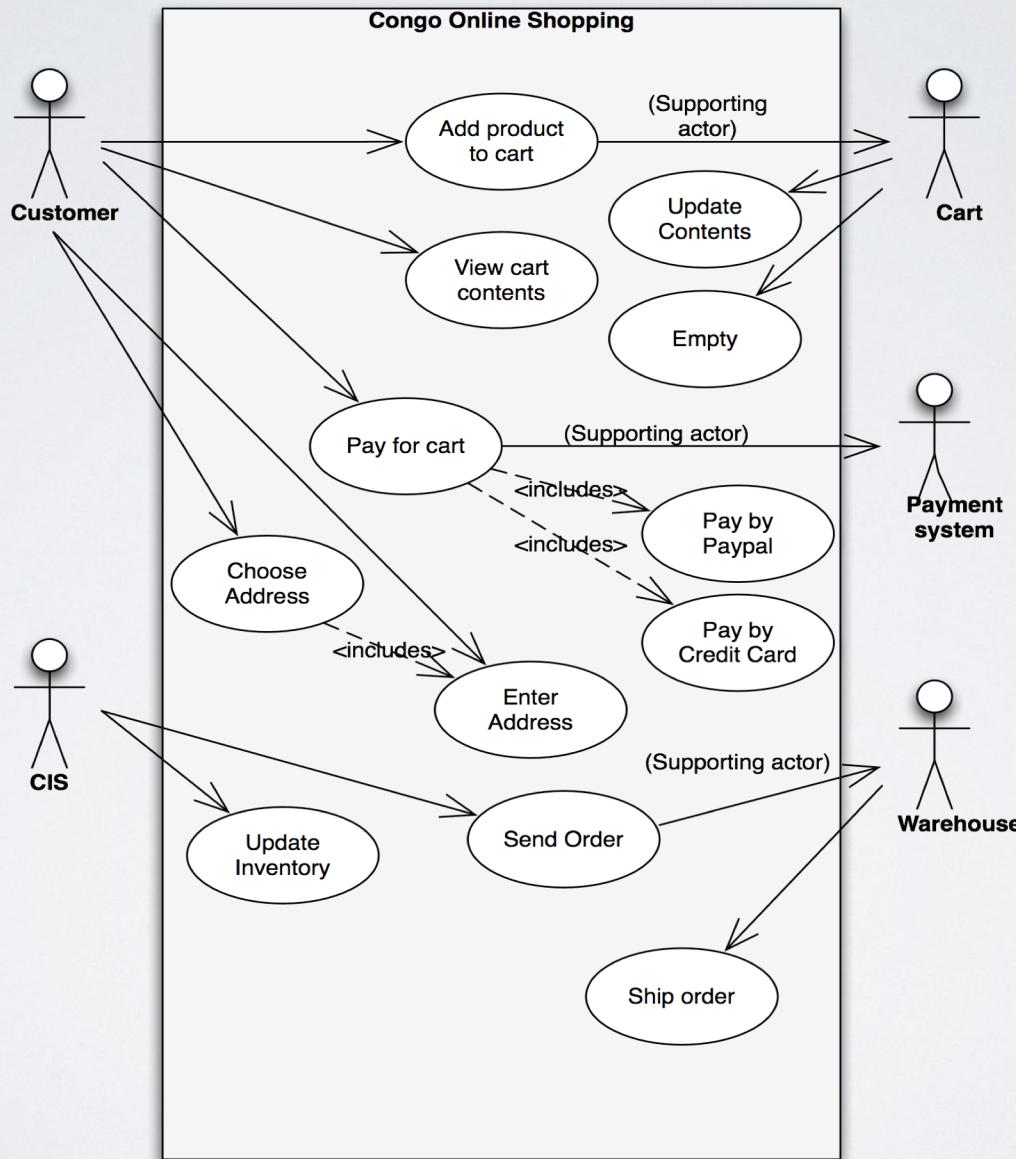
Looking back at the elicitation it turns out that they can pay through Paypal or Credit Card.

These are pretty different actions, so **we can split payment** so that it has **two additional sub cases** that can be included in the payment use case.

# Use cases: Update for complexity

Requirement	Use Cases
The customer will be able to add products to the shopping cart.	<i>Add product to cart (Customer)</i> <i>Update Contents (Cart)</i>
The customer will be able to view the contents of the shopping cart	<i>View cart contents (Customer)</i>
The customer will be able to pay for the contents of their shopping cart.	<i>Pay for cart (Customer)</i> <b><i>Pay by Paypal (Customer)</i></b> <b><i>Pay by Credit Card (Customer)</i></b> <i>Empty (Cart)</i>
The customer will be able to choose an address to ship the products.	<i>Choose address (Customer)</i> <i>Enter Address (Customer)</i>
The CIS inventory will be updated.	<i>Update inventory (CIS)</i>
The Warehouse will ship the order within 3 days of the order.	<i>Send order (CIS)</i> <i>Ship order (Warehouse)</i>

# USE CASES: RESULTING DIAGRAM



# ANALYSIS: EXAMINE SCENARIOS

Turning the analysis of complexity on its head, are there situations where one higher level use case could provide us with a view of a number of use cases?

Example:

- A series of steps for placing an order could be grouped in a scenario called “**Place order**” for the customer.
- It involves viewing the cart, choosing (or entering) an address and paying for it.

# ANALYSIS: EXAMINE SCENARIOS

- Scenarios provide us with a number of pieces of information:
  - Control flow (when a things sent from one actor to another – good for checking triggers and pre-conditions/post-conditions)
  - More actors
  - Shared use cases (things performed over and over)
  - Uncovering assumptions

# ANALYSIS: EXAMINE ASSUMPTIONS

By far the hardest part is trying to pick out those things that are being assumed in use cases and existing requirements.

Some are **functional**:

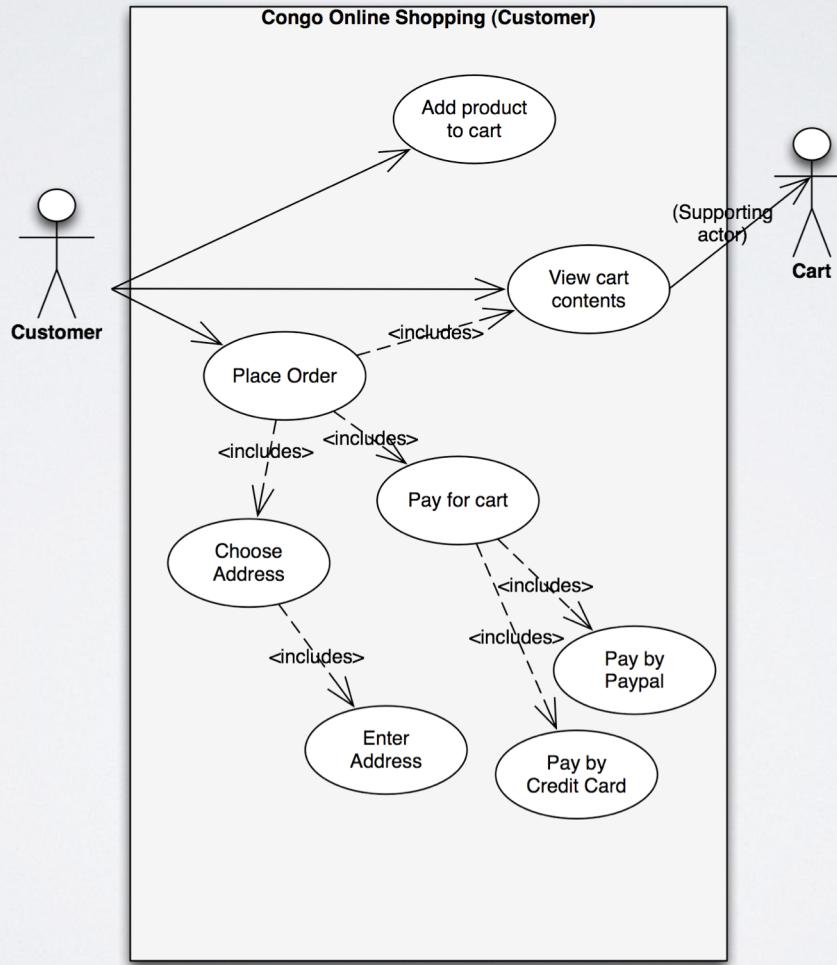
- Can the user view details about a product?
- Can the customer check the status of their order?
- Often these are **non-functional**.
  - Customers expect deliveries to be confirmed on arrival.
  - Customers expect their payments to be secure.
  - CIS needs to be updated within hours.

The only way to find these is to look closely at the sources of data where the requirements came from.

# Use cases: Update for assumptions

Requirement	Use Cases
The customer will be able to add products to the shopping cart.	<i>Add product to cart (Customer)</i> <i>Update Contents (Cart)</i> <b><i>View Item (Customer)</i></b>
The customer will be able to view the contents of the shopping cart	<i>View cart contents (Customer)</i>
The customer will be able to pay for the contents of their shopping cart.	<i>Pay for cart (Customer)</i> <i>Pay by Paypal (Customer)</i> <i>Pay by Credit Card (Customer)</i> <i>Empty (Cart)</i> <b><i>Place order (Customer)</i></b>
The customer will be able to choose an address to ship the products.	<i>Choose address (Customer)</i> <i>Enter Address (Customer)</i>
The CIS inventory will be updated.	<i>Update inventory (CIS)</i>
The Warehouse will ship the order within 3 days of the order.	<i>Send order (CIS)</i> <i>Ship order (Warehouse)</i> <b><i>Get confirmation (Customer)</i></b> <b><i>Check order status (Customer)</i></b>

# USE CASES: CONGO CUSTOMER



# SPECIFICATION

- With the analysis started, we can go back and update some of our requirements.
  - We can add requirements.
  - We can add sub requirements
  - We can add some of the non-functional aspects.

# ONLINE SHOPPING: REVISED REQUIREMENTS (INCOMPLETE)

1. The Customer will be able to view the details of a specific product.
2. The Customer will be able to add products to the shopping cart.
3. The Shopping cart will update its contents to include items added by the Customer.
4. The Customer will be able to view the contents of the shopping cart.
5. The Customer will be able to pay for the contents of their shopping cart.
6. The Customer will be able to pay for the contents of the shopping cart by Paypal or by Credit card.
7. The Customer will be able to enter an address to which the products should be shipped.
8. The Customer will be able to choose from a selection of addresses to which products can be shipped.
9. The CIS inventory will be update the inventory by removing the items from the order.
10. The warehouse will ship the order within 3 days of the order.

# SUMMARY

- Analysis tools such as use cases are ways to refine requirements.
- Much of the analysis comes from looking at your models, your data and your requirements and asking questions.
- Only practice will allow you to answer the questions well.
- Analysis does not stop with one iteration, often you will need many different tools to refine your requirements.

# SUMMARY

Scenarios are rich but unspecific

Code is **very** specific

What's the route?

# SUMMARY

Norman's Human-Action cycle

Goals

Intentions, specification, execution

Perception, interpretation, evaluation

Interrogate a scenario

Exhaust claims

# SUMMARY

World is messy

Systems are neat

UCD Problem

Need to throw out messiness

Need to keep realism

# READING

- Use cases on Wikipedia
- Benyon & McCaulay (2002) Scenarios and the HCI-Software Engineering design problem
- Sutcliffe (2003) Symbiosis & Synergy

# READING

- Interaction Design – Preece, Rogers and Sharp
- Chapter 10.6
- Scenarios and Design Cognition – Carroll
- <http://dx.doi.org/10.1109/ICRE.2002.1048498>
- • Rosson, M.-B. and Carroll, J.M. (2002). Usability engineering. San Francisco: Morgan Kaufmann!
- • Scenario Based Design – Chapter of the Handbook of HCI – posted on website!