Conceptual Models

A Brief Overview

Conceptual Models

Why are we talking about conceptual models?

Conceptual Models

Why are we talking about conceptual models?

- Storyboards, ideation, and paper prototyping
- Affordances, signifiers, and execution/evaluation
- Visual perception and gestalt principles
- Information design (language, vocabulary)
- Memory and attention

- User interface architecture
- State machines and state charts
- Graphical primitives and aggregates
- User interface design patterns (observer, state)
- Input, events, and callbacks

Conceptual Models

Conceptual model:

A representation of the designers view of how a user interprets a system

An idealized view of how a user should interpret a system

User

Understanding

Conceptual Model

Designer

Interpretation
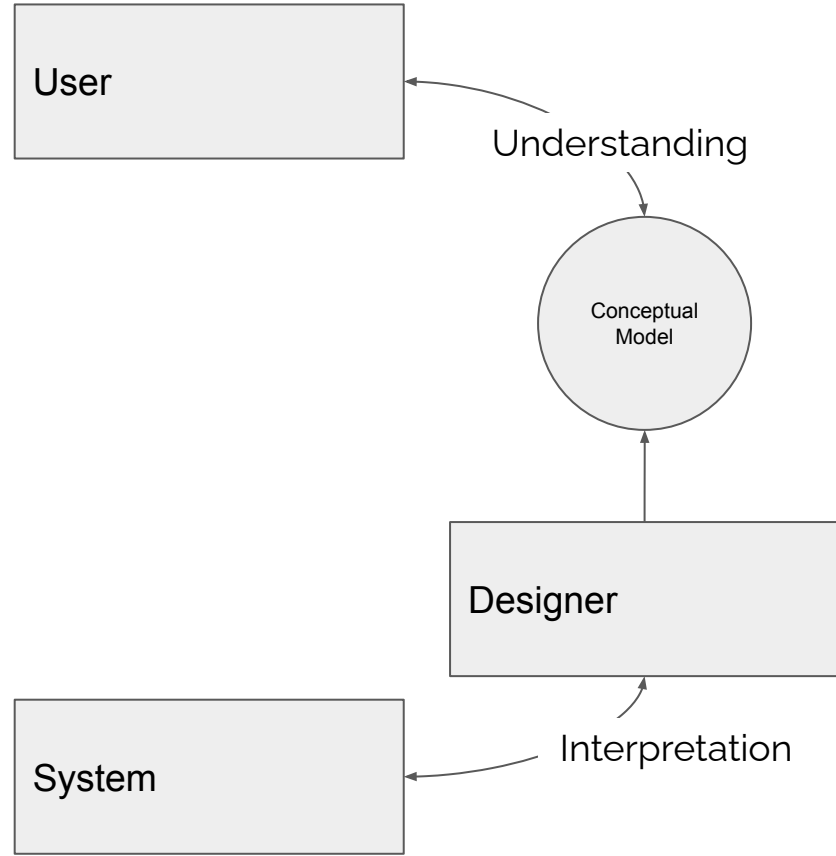
System

Conceptual Models

Conceptual model provides:

- A blueprint of a system
- Hierarchical representation
- Vocabulary

Conceptual Models

Creating a Conceptual model:

- Description of high-level functionality
- Objects, Operations, and Attributes
- Vocabulary
- Task Map

Conceptual Models

A simple example (_{adapted from} [Source] ₁):

The alarm clock

https://www.nytimes.com/wirecutter/reviews/best-alarm-clock/

Conceptual Models

The alarm clock:

- What is the form factor?
- What visual aesthetic should it follow?
- How many buttons and displays should it have?
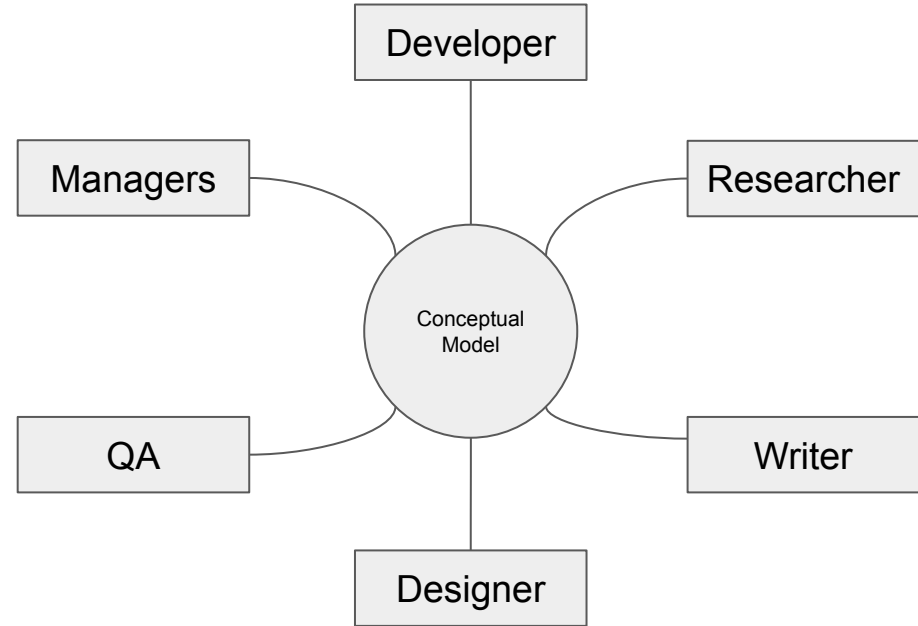- Where should interactors be located?

→

- Sketches
- Storyboards
- Prototypes
- etc

Conceptual Models

Iterative Development Cycle:

- Identifying and understanding user needs
- Functional requirements
- Design and evaluation
- Prototypes
- Product implementation
- Documentation
- Meetings of all kinds with a multidisciplinary team

Conceptual Models

The alarm clock requirements:

- Only one alarm time
- Only one alarm sound
- Displays current time
- The ability to set current time
- The ability to set alarm time
- The ability to turn off alarm

Conceptual Models

The alarm clock concepts:

- The clock stores the current time of day, continually updated as time passes
- The current time is always displayed
- The user can set the current time
- The user can set the alarm time
- If an alarm is set, it is triggered when current time equals alarm time
- The user can turn the alarm off if it has been triggered

Conceptual Models

Objects, Operations, and Attributes

Objects: The *things* in a system that a user can act upon

Operations: The *actions* that a user can carry out on an *object*

Attributes: The information required to complete an operation

Conceptual Models

Objects, Operations, and Attributes for the alarm clock:

Objects: **clock**, **alarm**

Operations:

- **Clock** supports viewing time, setting time
- **Alarm** supports setting current time, triggering alarm, and turning alarm off

Attributes:

- Current time, alarm time, and alarm status (*e.g.,* on or off)

Conceptual Models

Relationship Mapping

Objects: **clock**, **alarm**

Operations:

- **Clock** supports viewing time, setting time
- **Alarm** supports setting current time, triggering alarm, and turning alarm off

Attributes:

- Current time, alarm time, and alarm status (*e.g.,* on or off)

Conceptual Models

Relationship Mapping

Clock time == Alarm time

Objects:

➔ Time
   ◆ Clock
   ◆ Alarm

Conceptual Models

Vocabulary:

Clock: The current time, viewed by the user

Alarm: The time at which the alarm should be triggered

Alarm Status: Defines whether an alarm has been set or triggered

Time Set: An interface that allows the user to change the current clock time

Alarm Set: An interface that allows the user to change the alarm time

And so on.

Conceptual Models

A conceptual model is:

- Design metaphors
- Concepts exposed by the system to the user
- The relationship between concepts
- The mappings between concepts

A conceptual model is not:

- Visual design or UI
- Interactions (*e.g.,* mouse, keyboard, or touch input)
- Mental models
- Metaphors

Conceptual Models

A few guidelines:

- Keep the conceptual model as simple as possible
- Keep the conceptual model mapped as closely to the tasks you expect a user to complete as possible
  - The operations of your model should match the operations a user will carry out

Conceptual Models Summary

The conceptual model represents:

- How the designer interprets system operation:
  - High-level interpretation of a system that represents how the designer wants the user to think about that system
  - The user tasks supported by the system
- How the user understands the designer's intent:
  - Relies on users natural familiarity with the system domain
  - Forms the foundation for the mental models that users will build while using the system
- How the system actually works
  - The organization and operation of the system
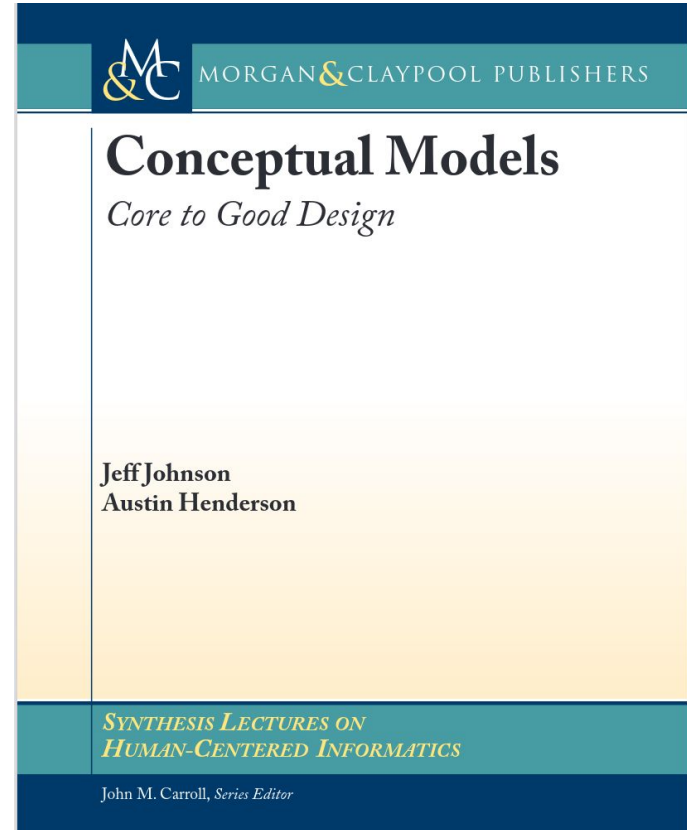
Conceptual Models

Getting Started:

- Pick an issue, assign a name and description
    - What issue(s) exist for a user when using a particular system?
- Describe existing user behavior for the issue
    - How does the user go about interacting with the system?
- Identify some examples of how a user executes tasks that lead to the issue
    - What are the user needs?
- Build your conceptual model using the information you have identified with the system

Conceptual Models

Conceptual Models: Core to Good Design

[PDF](#)

Conceptual Models

Core to Good Design

Jeff Johnson
Austin Henderson

References

1. Johnson, Jeff, and Austin Henderson. "Conceptual models: Core to good design." Synthesis Lectures on Human-Centered Informatics 4.2 (2011): 1-110.
2. Johnson, Jeff, and Austin Henderson. "Conceptual models: begin by designing what to design." interactions 9.1 (2002): 25-32.
3. Besnard, Denis, David Greathead, and Gordon Baxter. "When mental models go wrong: co-occurrences in dynamic, critical systems." International Journal of Human-Computer Studies 60.1 (2004): 117-128.
4. Kazi, Tarannum, et al. "Designer driving: drivers' conceptual models and level of trust in adaptive cruise control." (2007).
5. https://uxdesign.cc/start-your-designs-with-a-concept-7270e6b00fcc
6. https://uxdesign.cc/understanding-mental-and-conceptual-models-in-product-design-7d69de3cae26
7. Kurniawan, Sri. "Interaction design: Beyond human–computer interaction by Preece, Sharp and Rogers (2001), ISBN 0471492787." (2004): 289-289.