Parnas' Two Important Goals

- Design for change
  - Designers tend to concentrate on current needs
  - Special effort needed to anticipate likely changes
- Product families (now called: product lines)
  - Think of the current system under design as a member of a program family

Module

- A well-defined component of a software system
- A part of a system that provides a set of services to other modules
- A work assignment (Parnas)

Modular

- A complex system may be divided into simpler pieces called *modules*
- A system that is composed of modules is called *modular*
- Supports application of separation of concerns
    - when dealing with a specific module we want to be able to ignore details of other modules

Product Lines

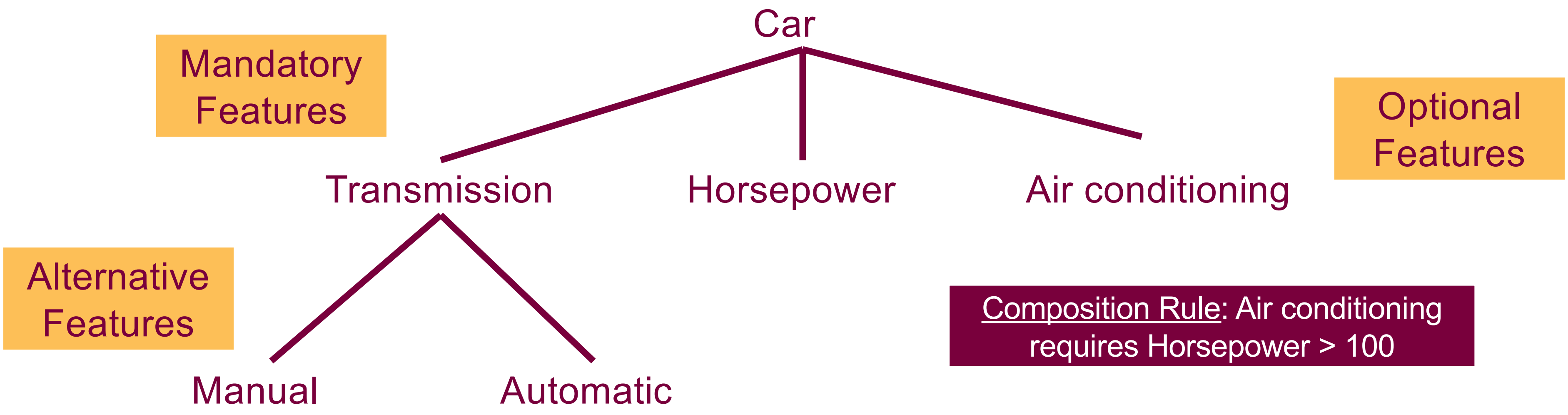
- Where modules are built by grouping related operations together, product lines take a more user focused approach
- A product line is an approach to development that identifies **Features** of the system in a top-down approach
  - A feature is originally defined as "*A prominent or distinctive user-visible aspect, quality, or characteristic of a software system or system*"*
    - In other words, a property of a system that is user interfacing or interactable
  - Instead of grouping functions together to form classes, identify features of the system that the user will interact with and develop functions to enable those features
  - Combine features together into a Feature Model
- A product line is formed when we take a feature model and apply it to multiple different products
  - The main purpose of a product line is to identify reusable portions of one product and reapply them across as many other product as possible
  - Increases robustness with respect to change and reduces repetition of work

*Kyo Kang, Sholom Cohen, James Hess, William Novak, and A Spencer Peterson.1990. Feature-oriented domain analysis (FODA) feasibility study. Software Engineering Institute. Universitas Carnegie Mellon, Pittsburgh, Pennsylvania (1990).

Feature Models

- Before creating Feature Models, need to first complete a domain analysis
- This process, unsurprisingly, is another name for the process of identifying requirements of the system and capturing knowledge of the system
- Then we begin the process of identifying and modelling features
    - Features need to be generic enough to be reusable, but specific enough to capture user interactions

Feature Model Diagram: Car Product Line



Car

Transmission

Horsepower

Air conditioning

Manual

Automatic

Mandatory Features

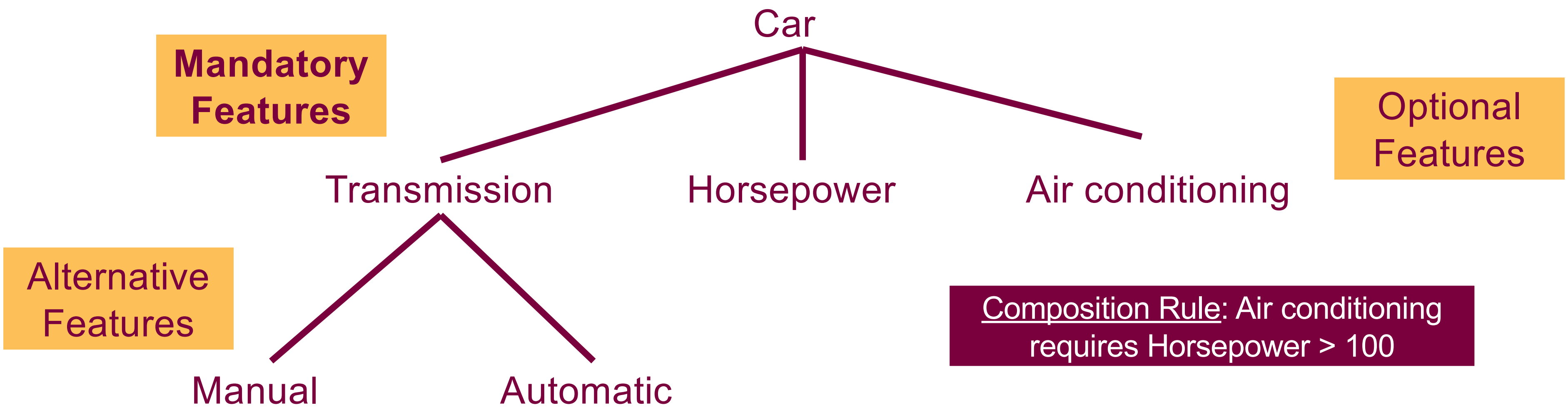
Optional Features

Alternative Features

Composition Rule: Air conditioning requires Horsepower > 100

Rationale: Manual more fuel efficient

Feature Model Diagram: Car Product Line

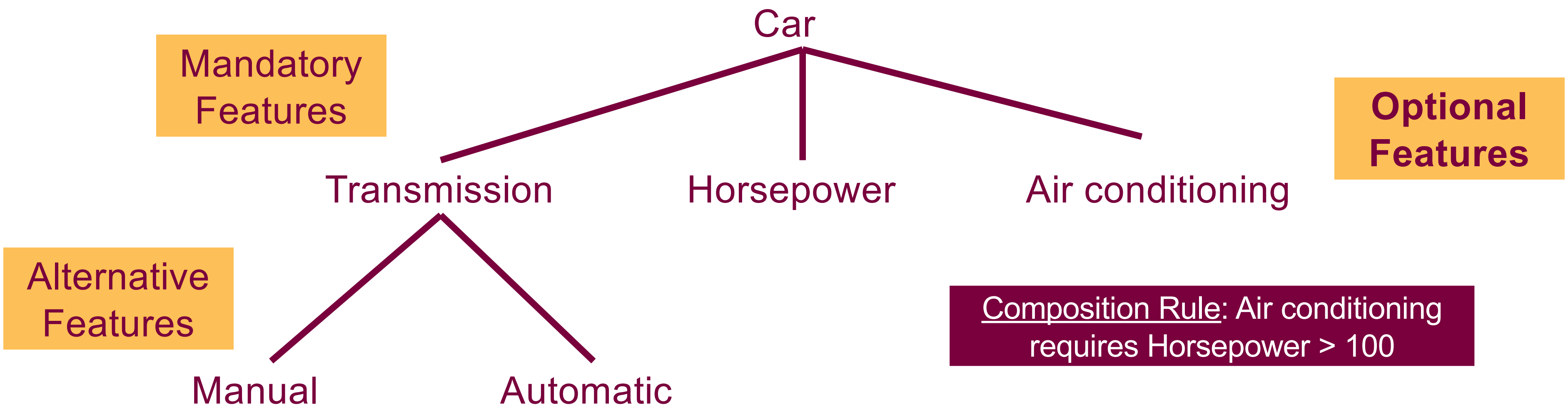


Composition Rule: Air conditioning requires Horsepower > 100

Rationale: Manual more fuel efficient

- **Mandatory features** are represented by a straight line with no decorators
- Think of these as an AND relationship when composed together for the product, in this case the “Car”

Feature Model Diagram: Car Product Line

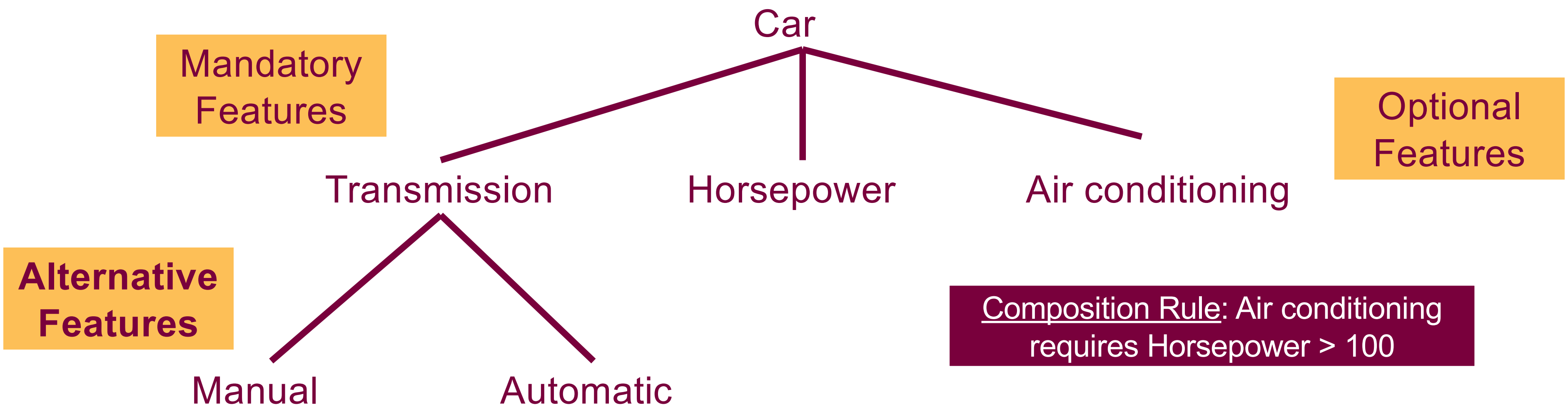


Composition Rule: Air conditioning requires Horsepower > 100

Rationale: Manual more fuel efficient

- **Optional features** are represented by the empty circle at the end
- Think of these as OR features for the product
- They do not need to be included for the completion of the product

Feature Model Diagram: Car Product Line



Car

Mandatory Features

Transmission

Horsepower

Air conditioning

Optional Features

Alternative Features

Manual

Automatic

Composition Rule: Air conditioning requires Horsepower > 100

Rationale: Manual more fuel efficient

- **Alternative features** are represented by the lines with an arc in between them and represent XOR relationships
- One of these features are mandatory for the completion of the product, but both cannot exist at the same time

What is front-end design?

- What does the system do?
  - How does the system convey its capabilities?
  - What can it do outside of its intended uses?
- Who/What are we designing for?
  - Does the design help people to accomplish a task?
  - Does the design invite people to use the product?
- This emphasizes people-focused design
  - Human-Computer Interaction!
  - One of the founders: Don Norman, has a great book on Design of Everyday Things

Design Explained

- At its core design is the same for front-end and back-end
  - The difference is in the ‘user’ of the design
- Designs boil down to how users will interact with a system or ‘thing’
  - These are the **Fundamental Principle of Interaction**
- The purpose of these fundamental principle is to outline the various portions of human psychology that are involved with design, especially when designing for people instead of other machines

Fundamental Principles of Interaction

- The 6 principles from Don Norman:
  1. Affordances
  2. Signifiers
  3. Conceptual System Model
  4. Mappings
  5. Feedback
  6. Constraints

1. Affordances and 2. Signifiers

- Affordances: what can a design do?
    - How does a design interact with the world around it?
    - What does its properties allow it to do?
- Signifiers: what does a design tell us it can do?
    - What does a design tell a user about it?
    - What does a design tell a user about its properties?
- The two are very similar and can often be confused and argued one way or another
    - When explaining a design distinction between affordances and signifiers are critical to understanding how the design works

3. The Conceptual Model

- This is heavily focused on the users understanding of the design
- An engineer that has never seen a class diagram before will have no idea what it means
- An end user that has never seen a bicycle before will have no idea what any of it means either
- The conceptual model is the understanding that a user will have of a design, how it relates to themselves, the environment, and the world at large

4. Mappings

- Leaning on a user's conceptual model, mappings are the relations that a user will make between the design, its environment, and the user themselves.
- If a user wants to turn on the lights in a room, they will map the switches to certain lights in the room
- If a user wants to turn a car, they will map the rotation of the steering wheel to a certain direction they want to turn in
- What about when mappings conflict?
  - When riding a motorcycle, in order to turn right users have to turn the handles left.

5. Feedback

- Feedback alerts the user to what actions they have taken
- These rely on the senses of the users
  - What are the most common forms of feedback you can think of?
- Often tied to satisfiability of a design
  - Feedback helps a users know when they have done something correctly
  - Also lets the user know when they have done something wrong
- Often, feedback can be represented by non-functional requirements of a system
  - Usability, look and feel, etc.

6. Constraints

- 4 kinds of constraints:
    - Physical
    - Cultural
    - Semantic
    - Logical
- Physical: the physical properties of a design that prevent certain actions
- Cultural: the influences of society in how we use certain designs
- Semantic: the meanings that we give to designs that are seemingly arbitrary
- Logical: the personal understanding that users make to create mappings between designs and the world

Design Decisions

- Requirements do not tell the full story…
  - They are open to interpretation in terms of how they are implemented.
  - Can be for user focused design and developer focused design
- Design decisions often require justification
  - Why is your system modular?
  - Why did you implement the requirements the way you did?
- Example: The Norman Doors problem
  - Req1: The door shall provide a barrier from outside elements (ie. Nature) and inside elements (ie. Lobby, home).
  - Req2: The door shall insulate inside temperature from outside temperature.
  - Req3: The door shall lock/unlock from outside with a locking/unlocking tool.
  - Req4: The door shall lock/unlock from inside with a locking/unlocking interface.
  - Req5: The door shall fit standard dimensions of 80in x 36in.

Design Decisions

How do you use these doors?

Design Decisions Justifications

- Why use a doorknob instead of a bar?
- Why use a key and lock instead of digital lock?
- Why use the color/material for the door?

Questions Summary: Design

- How to define the structure of a modular system?
- What are the desirable properties of that structure?
- What about the behavior of a modular system?
- How should a modular system interact with other systems/environments/users?
- How can we design systems with a focus on the people that will use them?
- How do we justify our design decisions?