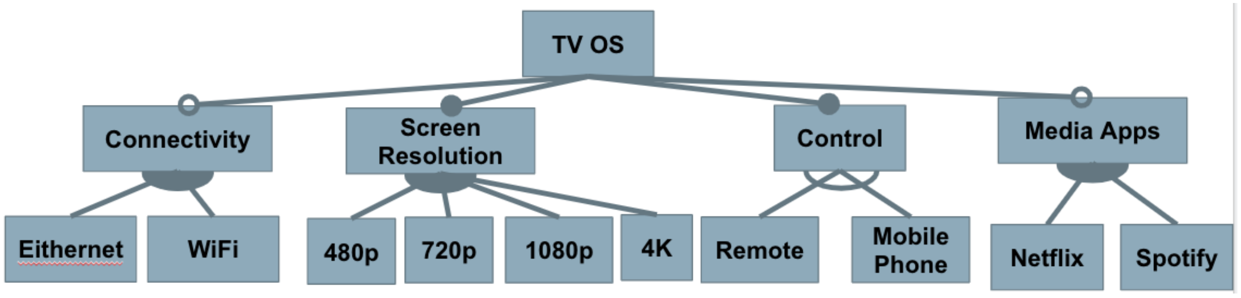
TDA 594/DIT 593 - Individual Written Assignment (Practice Version)

On all essay type questions, you will receive points based on the quality of the answer - not the quantity. You may either fill in this document or submit a separate document.

Question 1 - Domain and Application Engineering

Your company has developed a product line platform for smart TVs. Your current platform was developed according to the following feature diagram:



You receive the following requests from customers. For each request, decide if you will:

- **Extend the platform** to accommodate the request as one or more new features that can be reused in future products.
- **Add the new feature(s) to a single concrete application**, but not develop them for future reuse.
- **Decline the request** and continue to use your existing platform.

Write a short justification to explain your answer.

1. A game studio has requested you develop a new TV optimized to support 2K resolution (between 1080p and 4K) because that resolution offers a good balance between image sharpness and the speed the game can operate at.
2. Your current TVs come with a small selection of pre-installed apps (Netflix and Spotify currently). Customers have requested the addition of an app store where developers can publish their own apps and customers can download the apps that they want to use.
3. A restaurant chain has requested a special version of your TV that is locked to a single display for use in their stores for displaying menus, advertisements, and special videos. To extend your platform for their use, they ask for two features:
    a. A special app for that restaurant.
    b. A special remote containing only the power and volume buttons.

Question 2 - Feature Modelling

You work at a company that is developing a **word processor**. You have decided to develop this as a software product line, so that you can easily provide different feature sets for different types of customers.

1. Analyze the domain and identify a set of features.
2. Model the domain with a feature diagram.

Some items to consider:

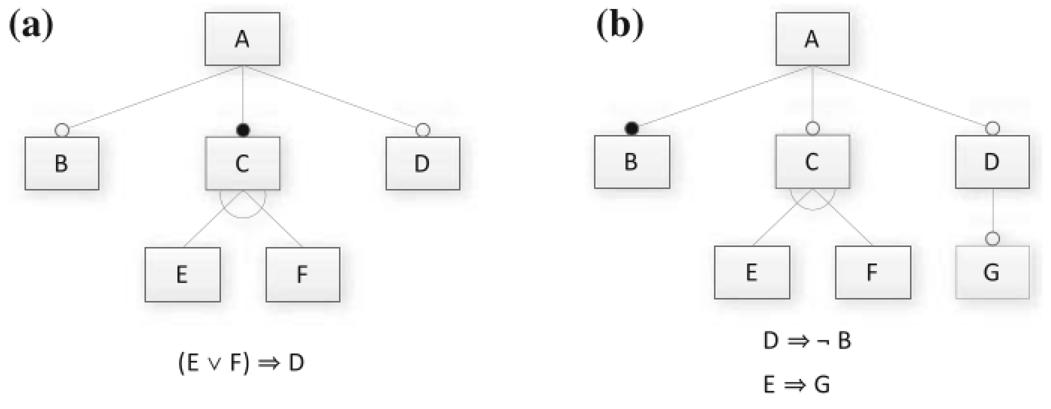
- Which features are likely to be requested by many customers?
- Which features are likely to be requested only by a few customers?
- Which features could distinguish your products from the products of your competitors in this market segment?
- Do not go crazy trying to identify all features. Try to capture an interesting set of important features (15 - 25, including all options for choices).
- Pay attention to feature dependencies and make sure you capture relevant cross-tree constraints and model structures (mandatory, optional, alternative, or).

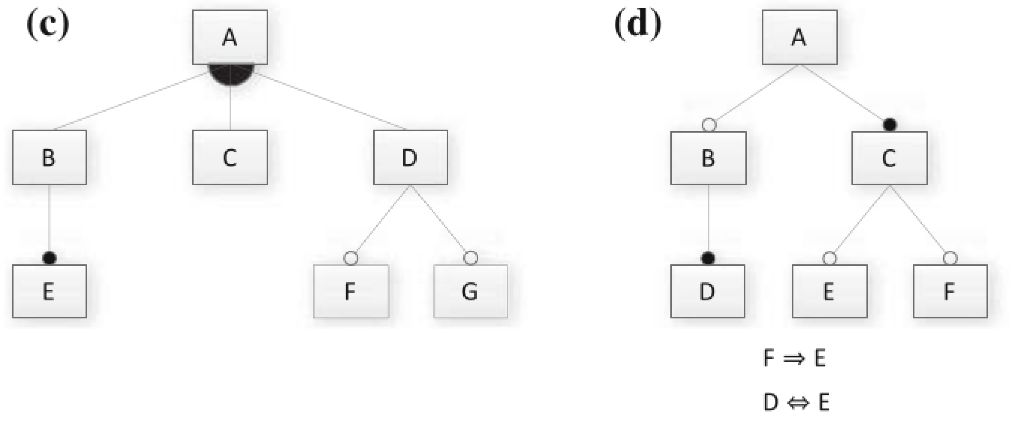
Question 3 - Model Analysis

Recall the following transformations from feature diagram to logic (where p and f are two features, and p is the parent of f):

- mandatory(p, f) ≡ $f \Leftrightarrow p$
  - filled circle: if parent is chosen, child must be chosen as well
- optional(p, f) ≡ $f \Rightarrow p$
  - empty circle: if parent is chosen, child may optionally be chosen
- alternative(p, $\{f_1,...,f_n\}$) ≡ $((f_1 \lor \dots \lor f_n) \Leftrightarrow p) \land_{(fi,fj)} \neg(f_i \land f_j)$
  - empty fan (XOR): if parent is chosen, exactly one child must be chosen
- or(p, $\{f_1,...,f_n\}$) ≡ $((f_1 \lor \dots \lor f_n) \Leftrightarrow p)$
  - filled fan (OR): if parent is chosen, at least one child must be chosen

Consider the following Feature Models:





1. Translate the feature model into a propositional logic formula. Note that the logical expressions next to models A, B, and D are cross-tree constraints that must be incorporated as well.
2. Provide two valid and two invalid feature selections (if possible).
3. Determine whether the feature model is consistent (are there any valid configurations?). If it is not consistent, identify one reason why.

Question 4 - Implementation Concepts

1. Consider compile-time and load-time binding of variability decisions.
   a. Define each and note how they differ from each other.
   b. Explain potential advantages and disadvantages of each.
2. Discuss which binding times (compile, load, run-time) are suitable (or ideal, or necessary) for the following features:
   a. Multiple alternative localization features (language selection, metric versus imperial units, and so forth) for the graphical user interface of a satellite navigation system.
   b. Two alternative sorting features in a database system: a very fast and a power-efficient sorting algorithm.
   c. Two alternative features in an operating system: single-processor support and multi-processor support.
   d. Two alternative features for edge representations in a library of graph algorithms: directed and undirected edges.
   e. Multiple optional features for supported file formats in printer firmware.

Question 5 - Design Patterns

In class, we discussed the following design patterns - strategy, decorator, factory, facade, adapter, and template method. Choose one of these patterns and:

1. Describe - in your own words - the goal of the pattern and how it is applied to a system.
2. Describe how this pattern would help enable controlled variability in an effective and efficient manner.
3. Give an example (not one that we covered in class) of a concrete system that would benefit from the application of this pattern. Draw a partial class diagram to help explain this example.

Note: We are not concerned with the exact syntax of this class diagram - we just want to see the relevant portion of the class layout of the proposed system.

Question 6 - Modularity and API Design

“Let’s Make a Deal” is a game where contestants are presented with three doors.

- One leads to a great prize, the others lead to nothing.
- Users select one door.
- Host opens one of the other doors.
- Users can then choose to open their door or the remaining unopened door.

You have been asked to implement Let’s Make a Deal as a web service. You must support:

- Creation of games.
- User selection of a door.
- The game will open one of the other doors.
- User opening of a door.
- Querying of the current state of the game and outcome (if complete) by user.
- Deletion of a game.

1. Create a REST API for this game. Determine the appropriate resources and verbs, and explain your API (what does applying a verb to that resource mean?).
2. Now, you want to extend your API into a generic, reusable API that could be used as the interface for additional games. Redesign your interface as a generic “game” interface, and explain why your new design could be reused for a different game.

Question 7 - System Testing (Category Partition Method)

You have created a utility intended to find all instances of a **pattern** in a **file**.

```
find(pattern,file)
```

This pattern can contain spaces and quotes. For example:

```
find(john,myFile)
```
Finds all instances of john in the file

```
find(“john smith”,myFile)
```
Finds all instances of john smith in the file

```
find(““john” smith”,myFile)
```
Finds all instances of “john” smith in the file

Use the category-partition method to identify a pool of valid test specifications.

1. Identify the **choices** that you control when testing.
2. For each choice, identify a set of **representative values** that could lead to different outcomes of the function.
3. Impose constraints on the choices to reduce the pool of test cases.
    a. **error** constraints identify values that should result in an error no matter what other values they are paired with.
    b. **single** constraints identity values that should result in normal execution, but should be tried once because they have the potential for error or strange behavior.
    c. **if-constraints** identify values that should only be used if other choices are set to particular values (“if choice X = THIS, then choice Y = THAT)

Question 8 - System Testing (Combinatorial Interaction Testing)

You are designing system-level tests for a web browser with multiple configuration options. You have extracted the following features, with the following value classes for each:

Allow Content to Load	**Notify About Pop-Ups**	**Allow Cookies**	**Warn About Add-Ons**	**Warn About Attack Sites**	**Warn About Forgeries**
Allow	Yes	Allow	Yes	Yes	Yes
Restrict	No	Restrict	No	No	No
Block		Block			

The full set of possible test specifications contains 144 options.

Create a covering array of specifications that covers all **pairwise value combinations** in fewer test specifications.

(hint: start with two variables with the most values and add additional variables one at a time)

Question 9 - Automation

Metaheuristic search techniques can be divided into local and global search techniques.

1. Define what a “local” search and a “global” search is.
2. Contrast the two approaches. What are the strengths and weaknesses of each?
3. Choose one search algorithm and briefly explain how it works. State whether it is a global or local search, and explain why it belongs to that category.

Question 10 - Research in Software Product Lines

Read the following research paper:

Wardah Mahmood, Daniel Strüber, Thorsten Berger, Ralf Lämmel, Mukelabai Mukelabai. *Seamless Variability Management With The Virtual Platform*. Available at https://arxiv.org/abs/2103.00437.

After reading this paper, explain (in your own words) the following:

1. What problem are the authors attempting to address?
2. Why is this problem important to address?
3. What did the authors do to address this problem?
4. What conclusions did they come to?
5. What is one thing you think could be done to extend this work in the future? Do not state one of the ideas for future work that the authors proposed themselves, but come up with your own idea.