# CSCC01 – Software Engineering

Software Requirements

Separating the Problem from the Solution

❑ A separate problem description is useful
  - It can be discussed with stakeholders
  - It can be used to evaluate design choices
  - It is a good source of test cases

❑ Still need to check
  - Solution correctly solves the stated problem (verification)
  - Problem statement corresponds to the needs of the stakeholders (validation)
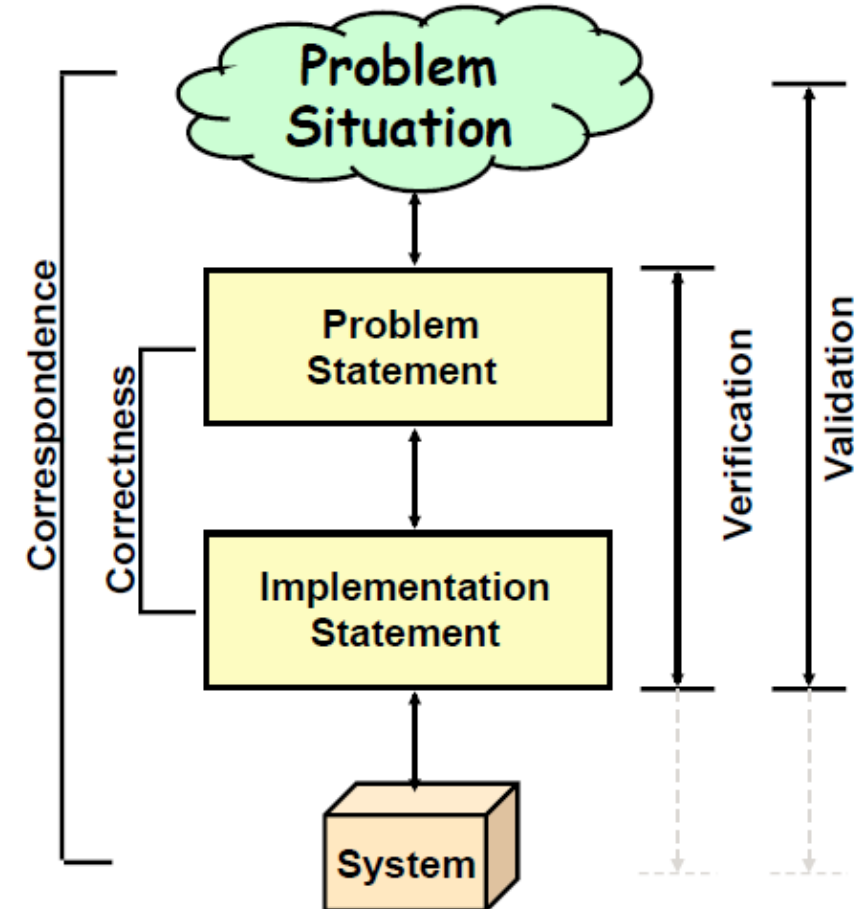
Stakeholders

❑ Stakeholder analysis

- Identify all the people who must be consulted during information acquisition

❑ Stakeholders include

- Users
- Customers
- Business analysts/marketing team
- Training and user support staff
- Technical authors
- Designers
- Project manager

Identifying Stakeholders' Goals

❑ Approach
- Focus on **why** a system is required
- Express the "why" as a set of stakeholder goals
- Use goal refinement to arrive at specific requirements
- Goal analysis: document, organize and classify goals
- Goal evolution: refine, elaborate, and operationalize goals
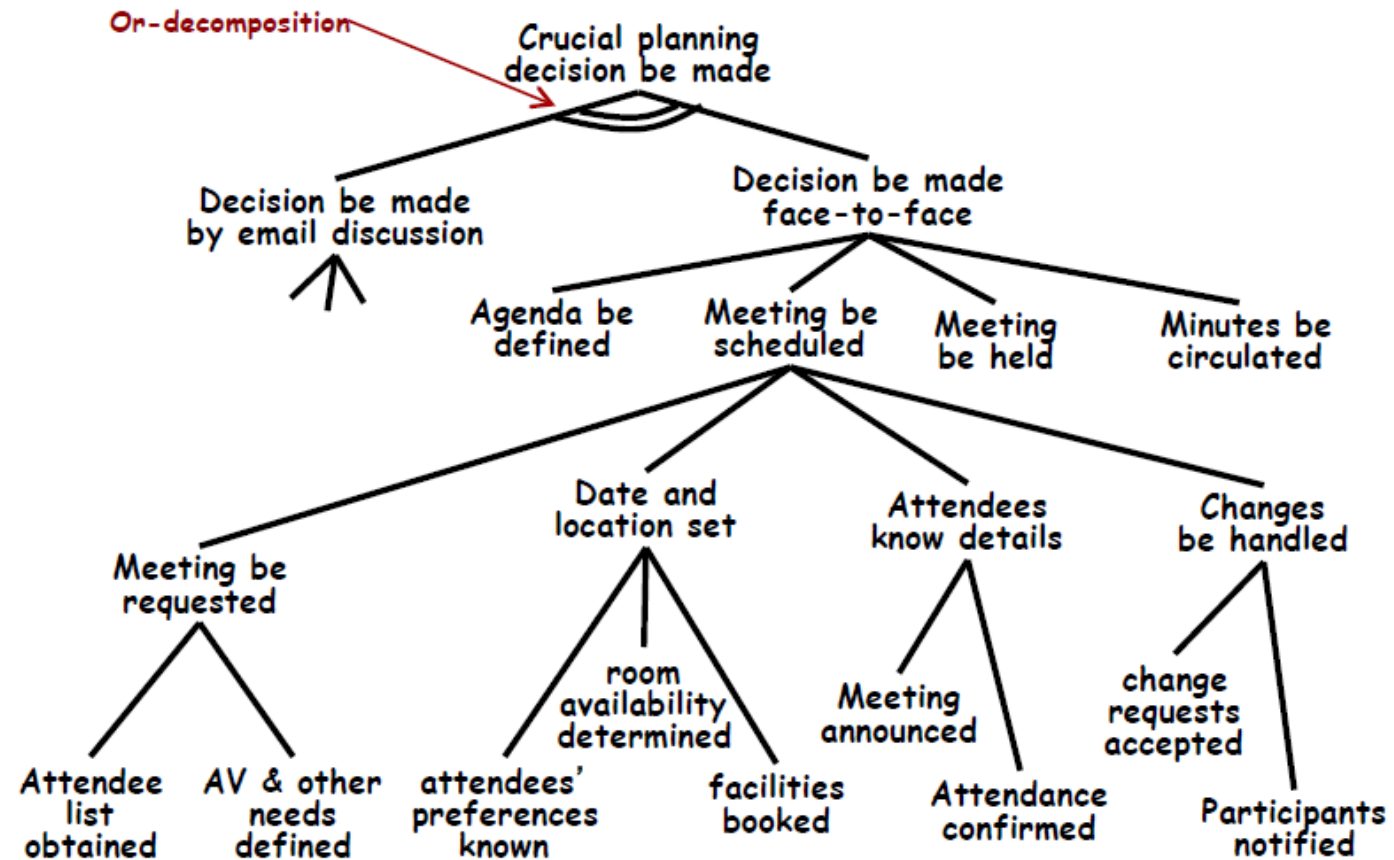- Goal hierarchies show refinements and alternatives

❑ Advantages
- Reasonably intuitive
- Explicit declaration of goals provides sound basis for conflict resolution

❑ Disadvantages
- Captures a static picture - what if goals change over time?
- Can regress forever up (or down) the goal hierarchy

Goal Elaboration

Goal Analysis

❑ Goal Elaboration
- "Why" questions explore higher goals (context)
- "How" questions explore lower goals (operations)
- "How else" questions explore alternatives

❑ Relationships between goals
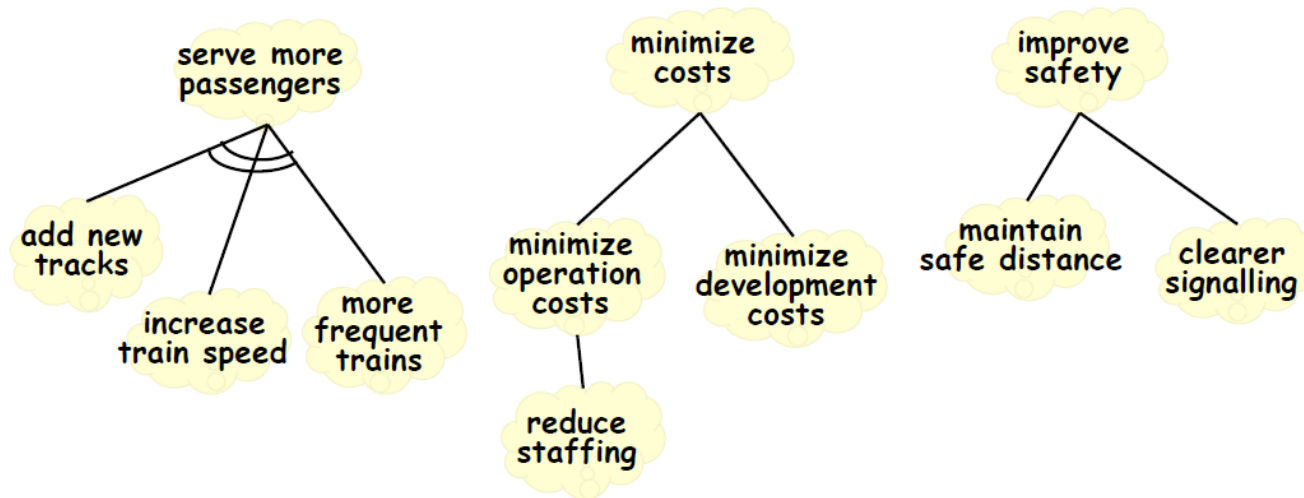- One goal helps achieve another (+)
- One goal hurts achievement of another (-)
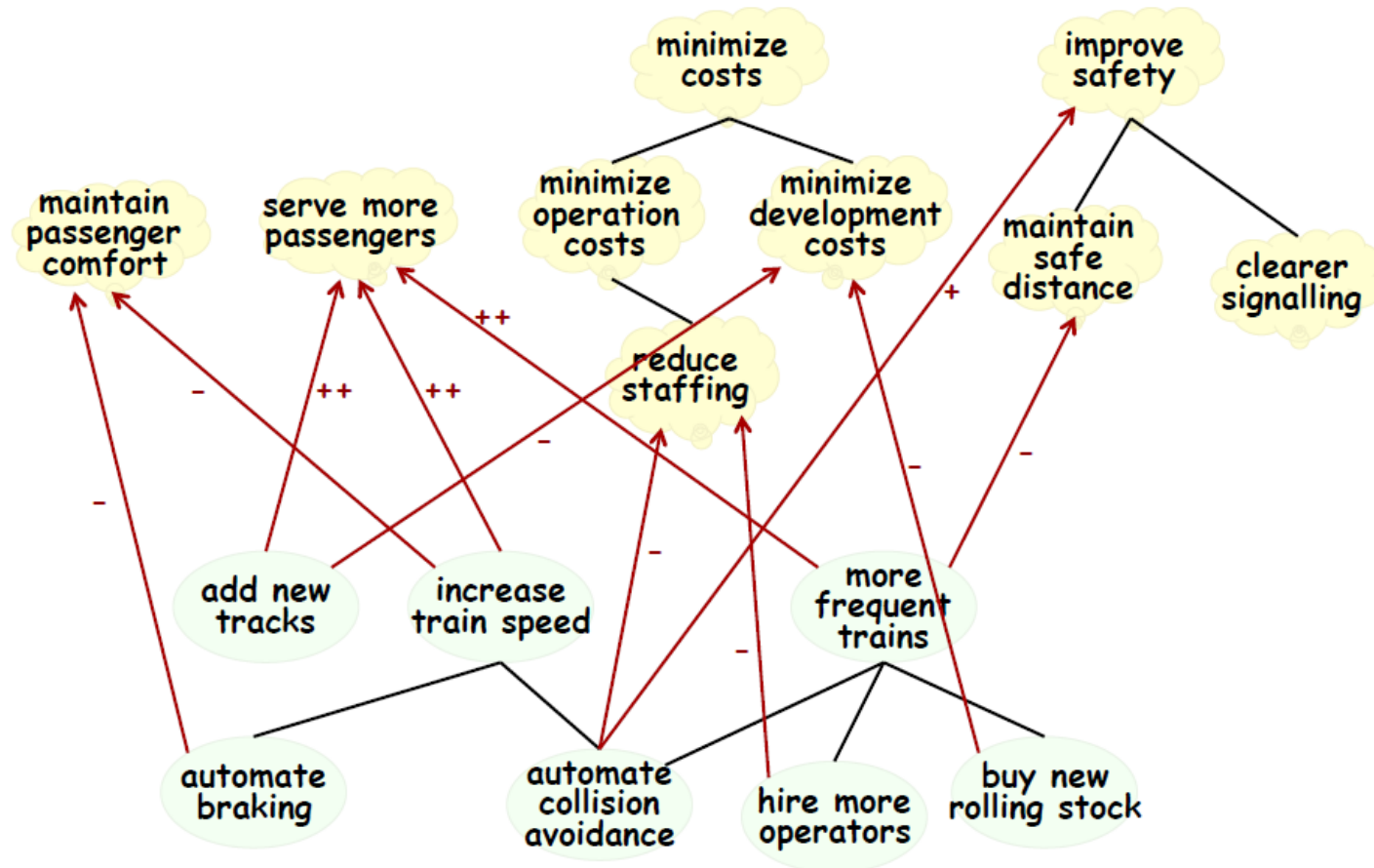- One goal makes another (++)
- One goal breaks another (--)
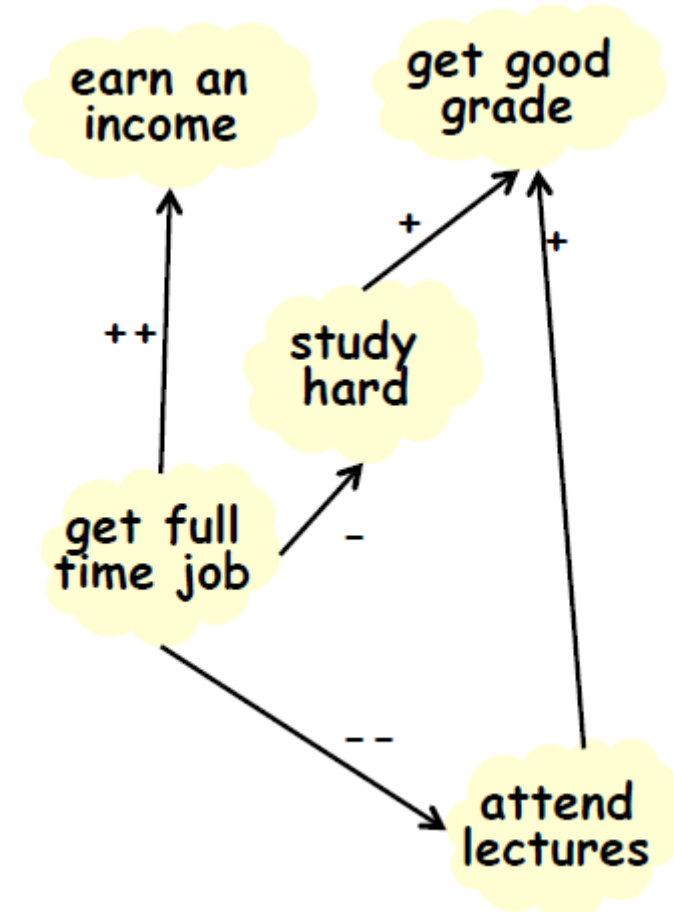
Soft Goals

❑ Some goals may be subjective
- Often related to qualitative aspects such as usability, performance, etc
- Treat these as *soft goals*
- E.g. "system be easy to use", "access be secure"

❑ Train System Example:

Soft Goals as Selection Criteria

From Requirements to Design

1. ## Identify the problem
    - What is the goal of the project?
    - What is the vision of those who are pushing for it?

2. ## Scope the problem
    - Given the vision, how much do we tackle?
    - What new functionality will be needed?

3. ## Identify solution scenarios
    - Given the problem, how will users interact with the software to solve it?

4. ## Map onto the architecture
    - How will the needed functionality be met?
    - What new modules / classes will be needed?

Identifying Actors

❑ Primary actors

- Who will need support from the system to do their daily tasks?
- Who or what has an interest in the results that the system produces ?

❑ Secondary actors

- Who will maintain, administrate, keep the system working?

Personas

❑   A persona is a fictional character representing a specific user type or group

❑   Involves detailed descriptions of user characteristics, goals, behaviors, and preferences
  - Imaginary names and backgrounds to make personas relatable

❑   Developed through research, surveys, and interviews

❑   Advantages
  - Enhances user-centered design and user experience
  - Facilitates communication within the development team

Key Distinctions

A requirement (goal)

reduce the number of spelling mistakes

Functions

identify complete word

highlight misspellings

suggest words from dictionary

pop up a menu

A Use Case

The user types text as usual. When the user completes each word, the system looks it up in the dictionary. If it is not in the dictionary, the word is underlined in red. The user can click on any underlined word, to see a popup menu of suggested alternatives. Clicking any of these alternatives causes it to replace the original word.

Domain Concepts

dictionary

space

suggestion

replacement

spelling

selection
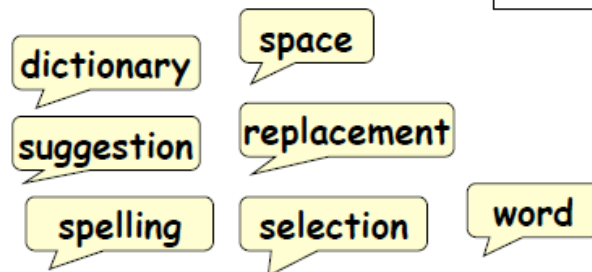
word

Domain Model
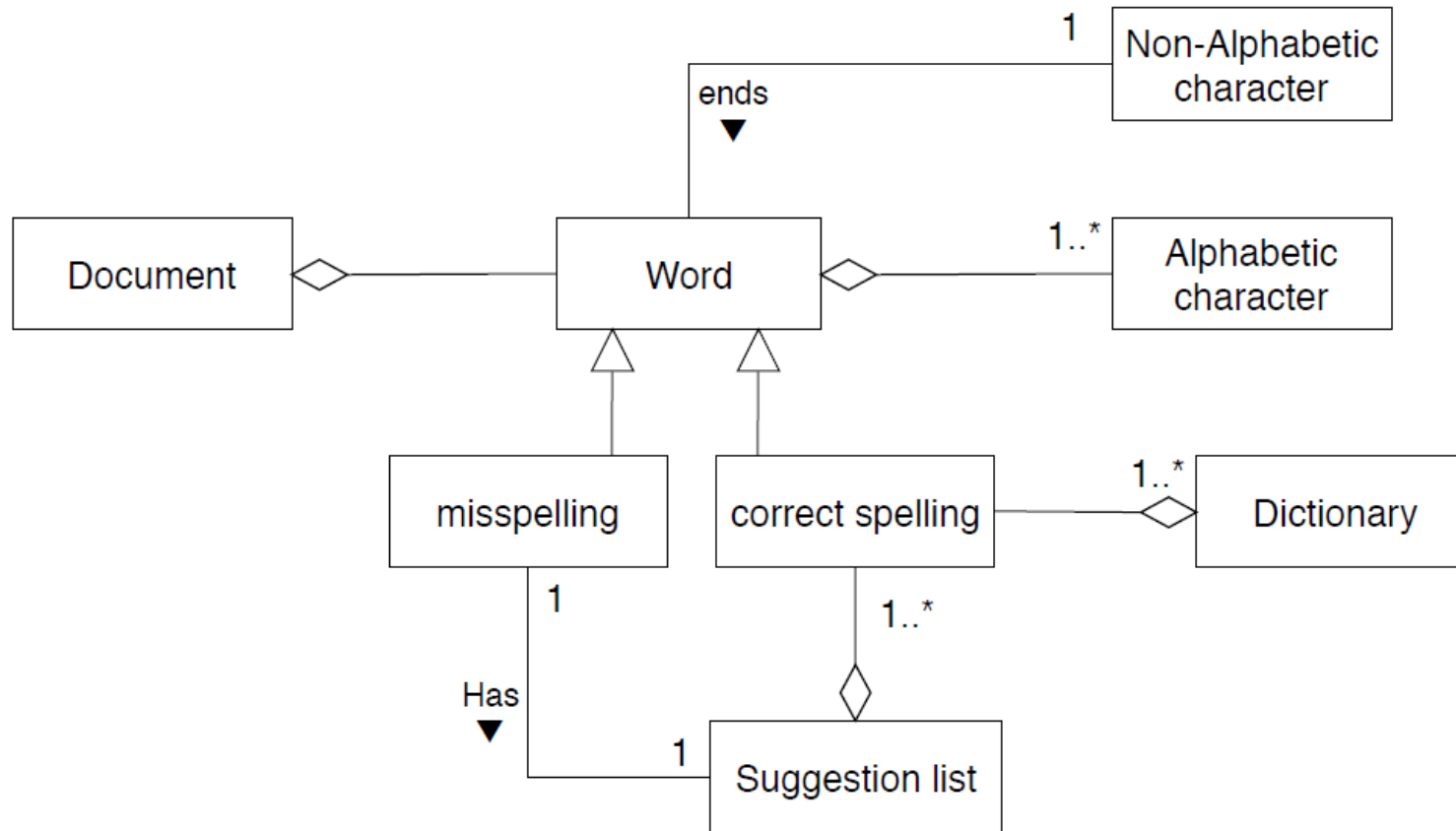
Exploring Goals
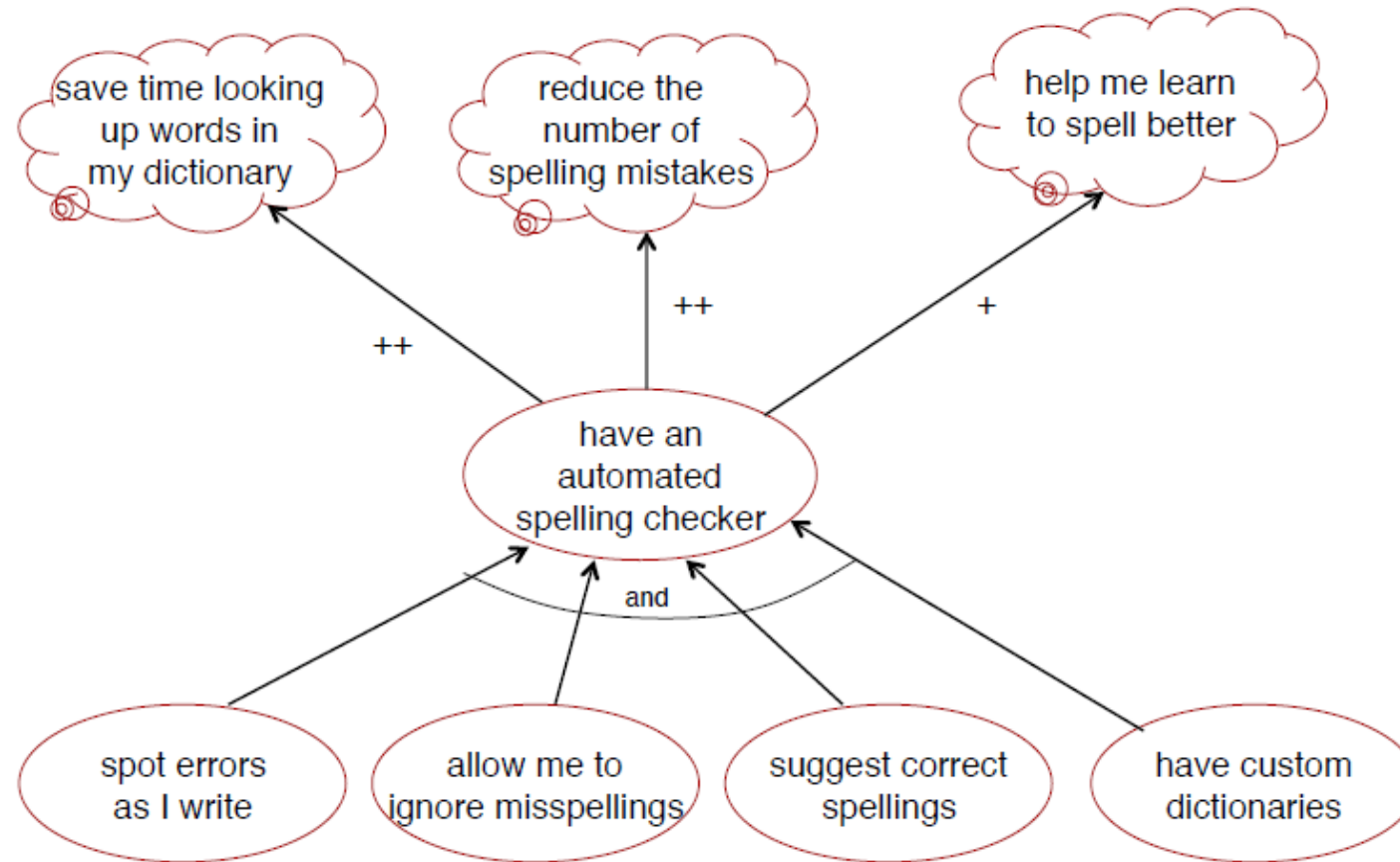
Some Requirements Emerge

❑ **Functional Requirements**

- User can see definitions for suggested spellings
- User shall be able to add custom dictionaries
- User shall be able to add new words to a custom dictionary
- User shall be able to declare certain words be ignored for spell checking for the current document

❑ **Quality Requirements**

- Checking and suggesting should be fast
- Highlighted misspellings must be clearly visible

Use Case Diagram

Sample Use Case Documentation

Brief name (This is all that appears on the on use case diagram)

Name: Place Order

Precondition: A valid user has logged into the system.

Description:

1. The use case starts when the customer selects Place Order.
2. The customer enters his or her name and address.
3. If the customer enters only the zip code, the system will supply the city & state.
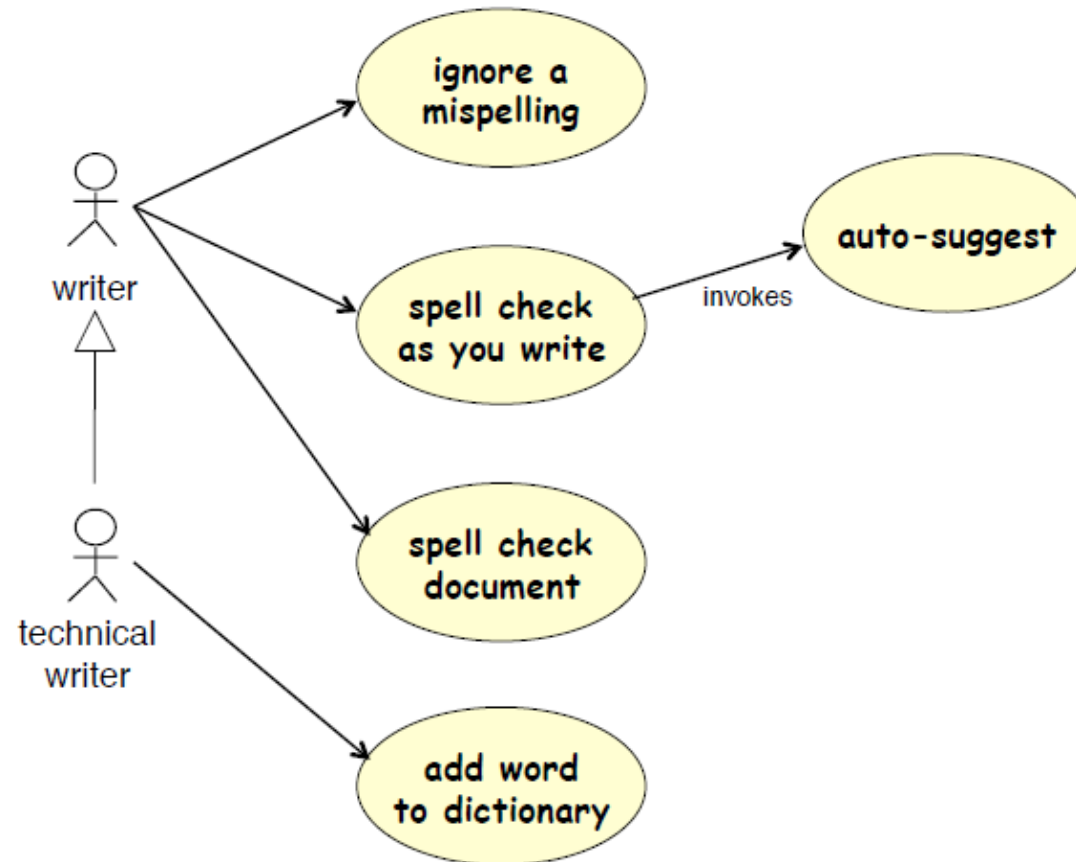4. The customer will enter product codes for the desired products.
5. The system will supply a product description and price for each item.
6. The system will keep a running total of items ordered as they are entered.
7. The customer will enter credit card payment information.
8. The customer will select Submit.
9. The system will verify the information, save the order as pending, and forward payment information to the accounting system.
10. When payment is confirmed, the order is marked Confirmed, an order ID is returned to the customer, and the use case ends.

Exceptions:

In step 9, if any information is incorrect, the system will prompt the customer to correct the information.

Postcondition: The order has been saved in the system and marked confirmed.

Clear start and end

Every step clearly identifies who carries it out

Clear statement of any effects