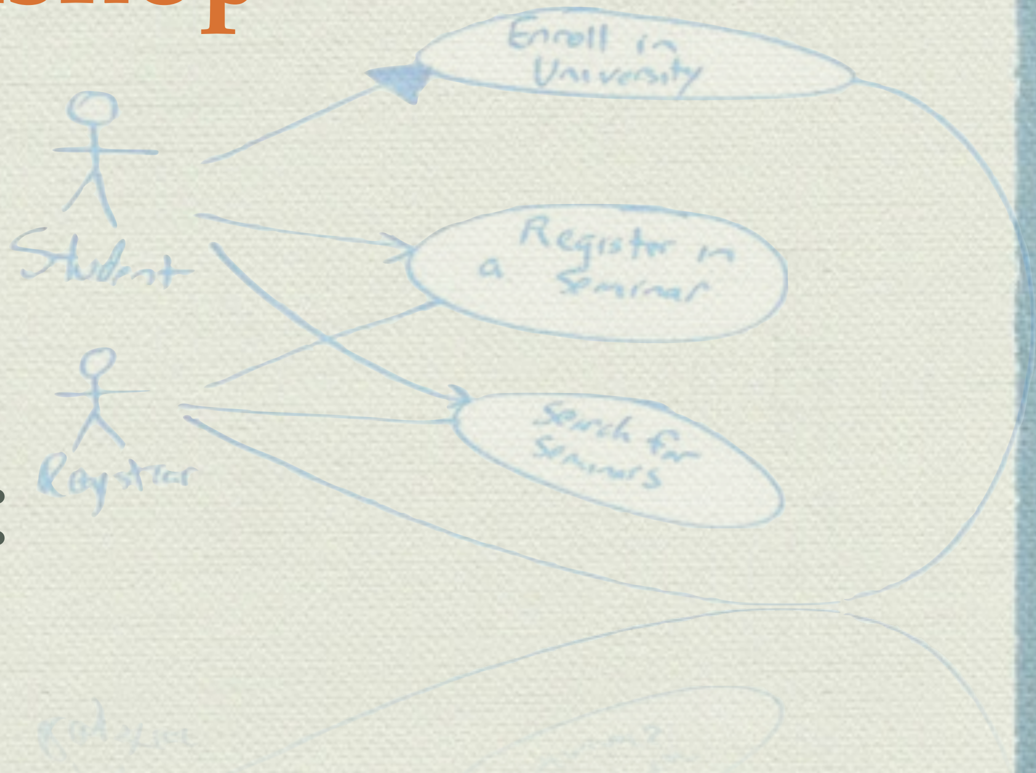
Software Engineering Workshop

Workshop 3

Working with Requirements: Use Case Diagrams (Part 2)



Slides prepared by Marwah Alaofi

Quick Review

- What purpose does the use case diagram serve?
- What graphical notations it involves?
- How useful it could be in the software development?

In today's workshop you'll learn..

- The <<include>> relationship
- The generalization relationship between use cases and between actors
- The <<extend>> relationship

Use Case Relationships

- When looking at your use case scenarios, you will notice that there is some similarity between steps in different use cases.
- You may also find that some use cases work in several different modes or special cases.
- You may also find a use case with multiple flows/scenarios throughout its execution, and it would be good to show those important optional cases on your use case diagrams

Use Case Relationships (Cont.)

- Wouldn't it be great if you could get rid of the repetition between use case descriptions and show important optional flows.
- You can show reusable, optional, and even specialized use case behavior between use cases.

Weblog Content Management System

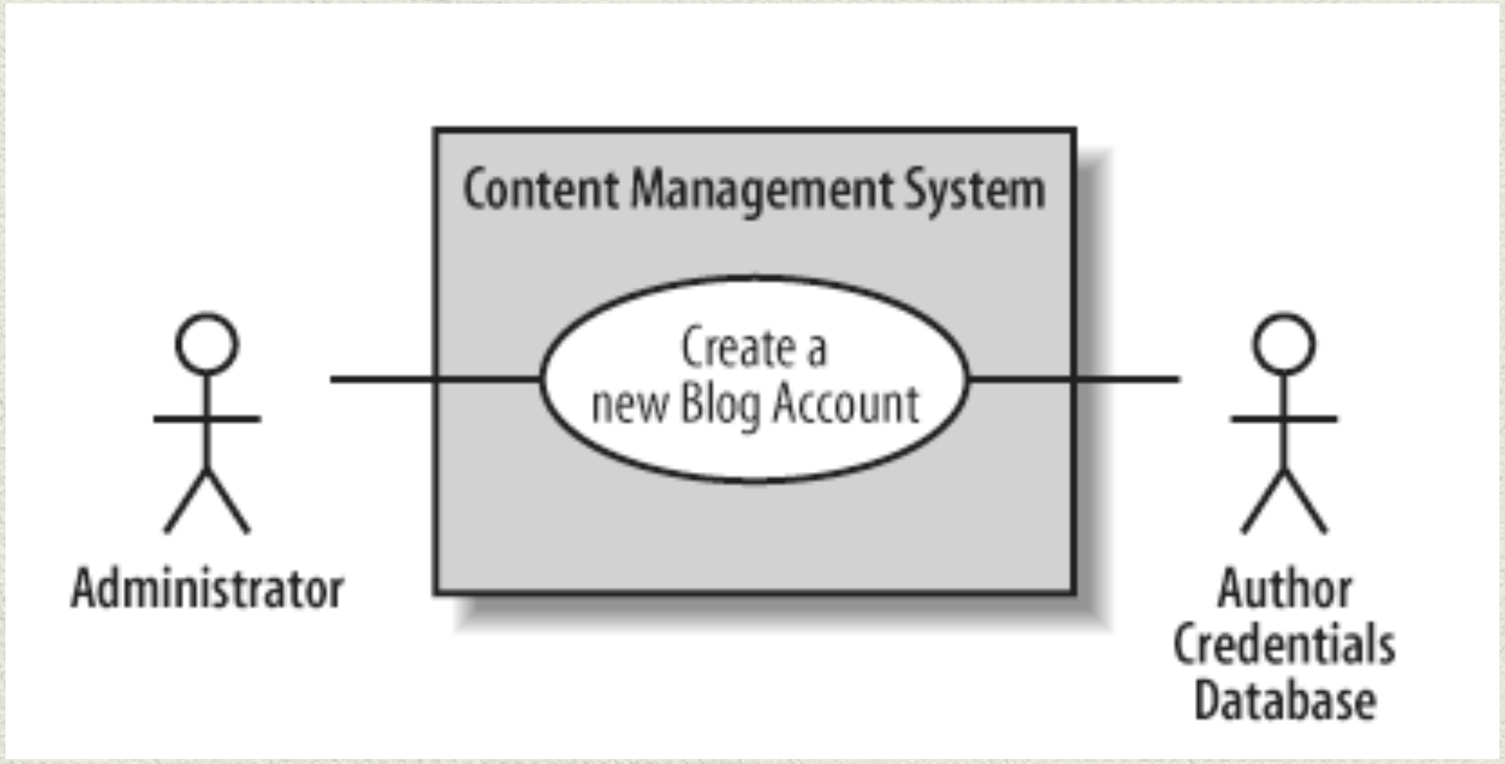
- Suppose we're defining requirements for a weblog content management system (CMS).

Weblogs, commonly referred to as blogs, originally started out as privately maintained web pages for authors to write about anything. These days, blogs are usually packaged into an overall CMS. Bloggers submit new entries to the system, administrators allocate blogging accounts, and the systems typically incorporate advanced features, such as RSS feeds. A well-publicized blog can attract thousands of readers (see O'Reilly's blogging site at http://weblogs.oreillynet.com).

Weblog Content Management System (Cont.)

REQUIREMENT A.1

The content management system shall allow an administrator to create a new blog account, provided the personal details of the new blogger are verified using the author credentials database.


Content Management System
Create a new Blog Account
Administrator
Author Credentials Database

“Create a new blog account” Use Case

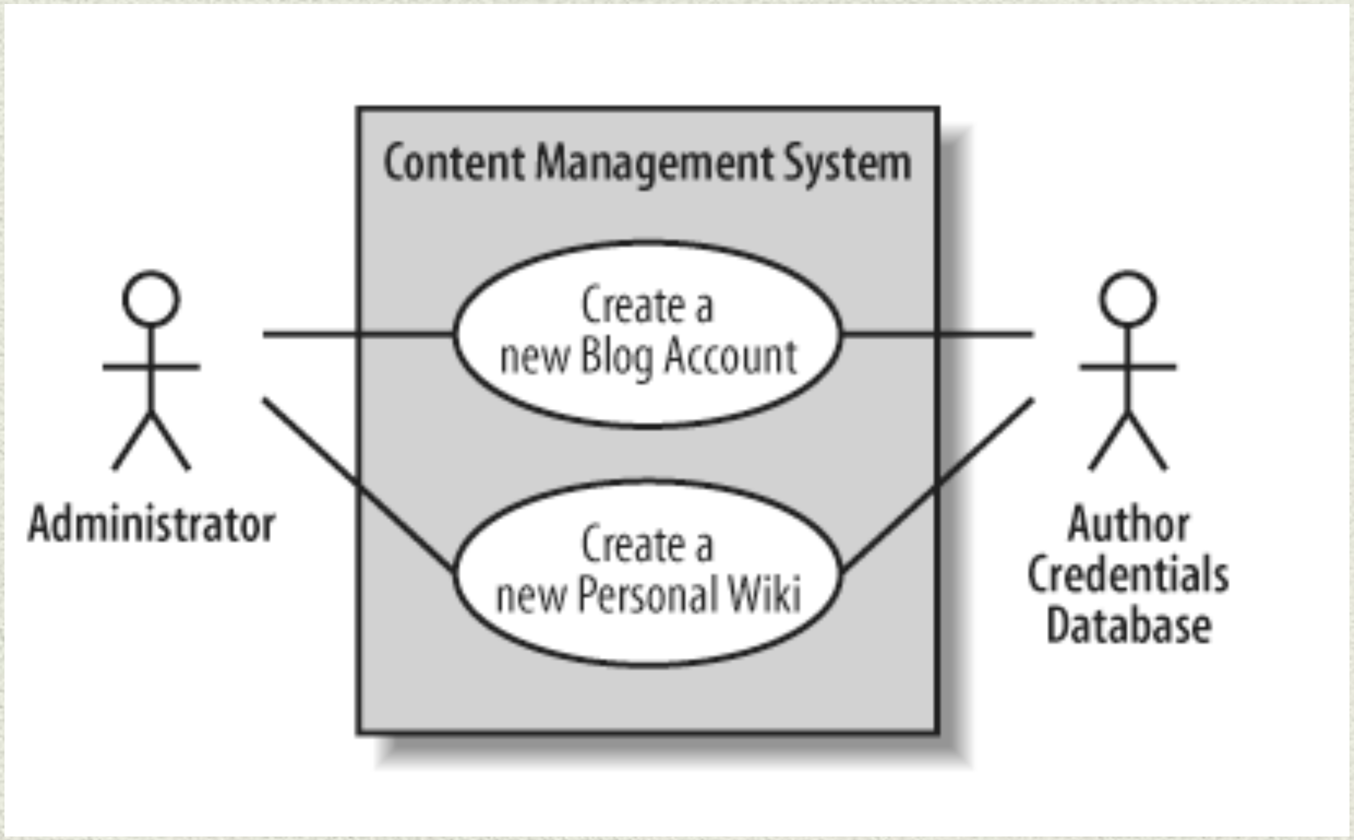
Create a new blog account

1. **The Administrator asks the system to create a new blog account.**
2. **The Administrator selects an account type.**
3. **The Administrator enters the author's details.**
4. **The author's details are verified using the Author Credentials Database.**
5. **The new blog account is created.**
6. **A summary of the new blog account's details are emailed to the author.**

Weblog Content Management System (Cont.)

REQUIREMENT A.2

The content management system shall allow an administrator to create a new personal Wiki, provided the personal details of the applying author are verified using the Author Credentials Database.


Content Management System
Create a new Blog Account
Create a new Personal Wiki
Administrator
Author Credentials Database

“Create a new personal wiki” Use Case

Create a new personal Wiki

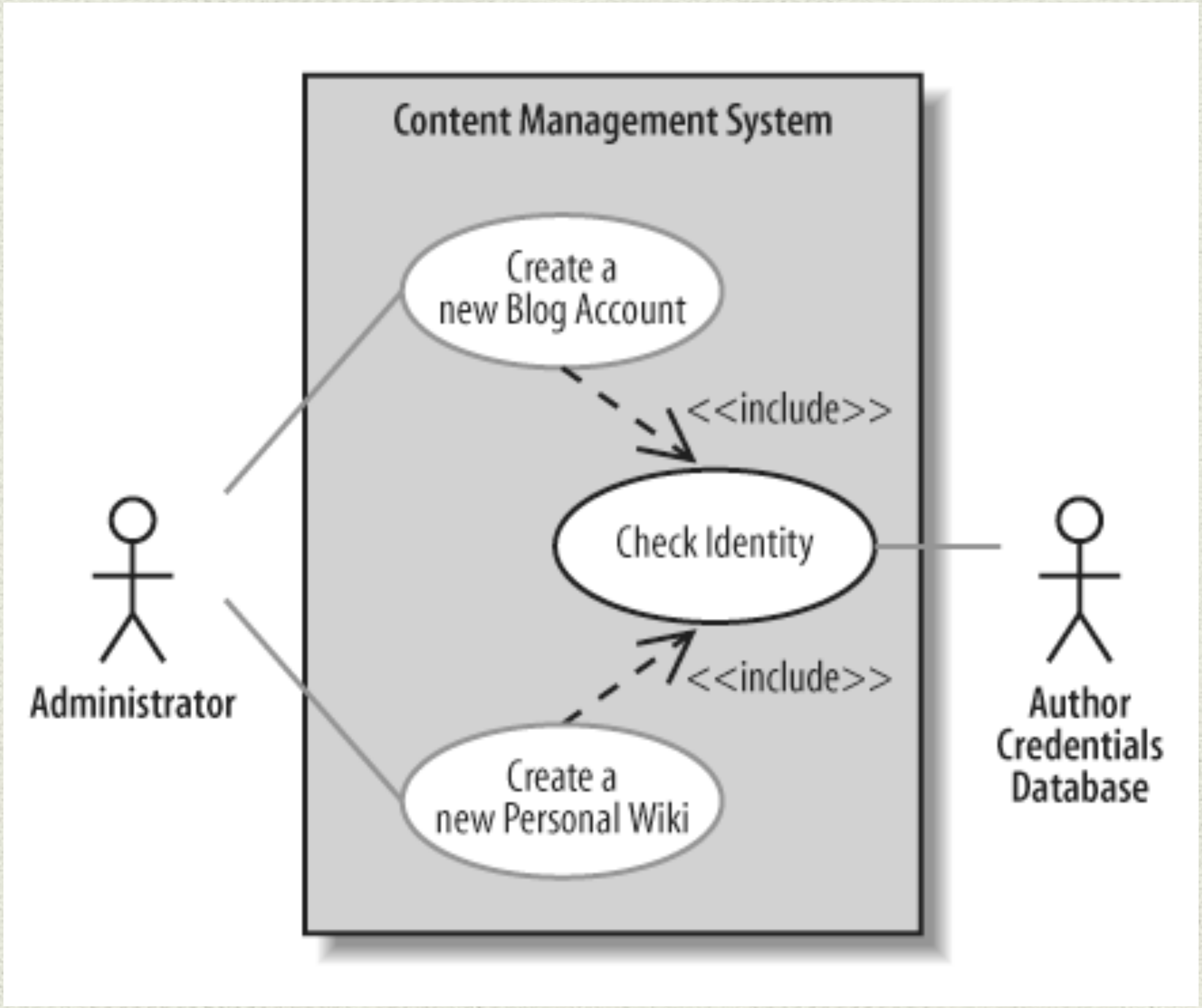
1. **The Administrator asks the system to create a new personal Wiki.**
2. **The Administrator enters the author's details.**
3. **The author's details are verified using the Author Credentials Database.**
4. **The new personal Wiki is created.**
5. **A summary of the new personal Wiki's details are emailed to the author.**

The <<include>> Relationship

- There is some redundancy between the two use case descriptions.
- Both Create a new Blog Account and Create a new Personal Wiki need to check the applicant's credentials.
- This behavior is simply repeated between the two use case descriptions.

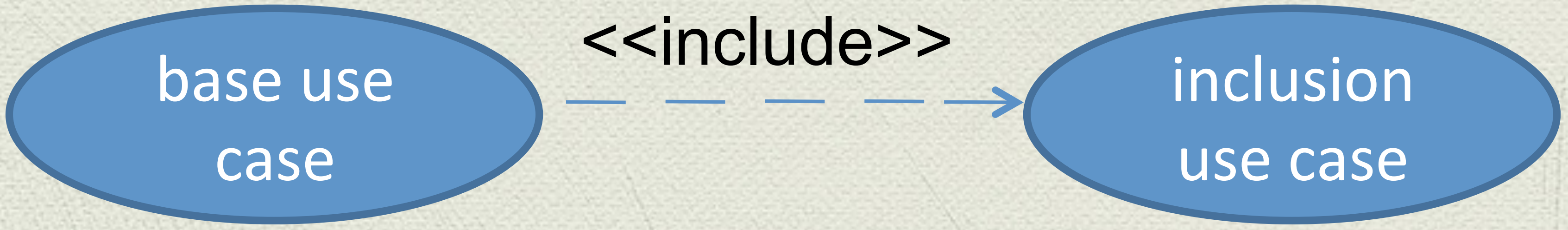
The <<include>> Relationship

- This repetitive behavior shared between two use cases is best separated and captured within a totally new use case.
- This new use case can then be reused by the `Create a new Blog Account` and `Create a new Personal Wiki` use cases using the `<<include>>` relationship.


Content Management System
Create a new Blog Account
<<include>>
Check Identity
<<include>>
Create a new Personal Wiki
Administrator
Author Credentials Database

The <<include>> Relationship

- An include relationship from one use case (called the base use case) to another use case (called the inclusion use case) indicates that the base use case will include or call the inclusion use case.
- An include relationship is shown as a dashed arrow from the base use case to the inclusion use case marked with the include keyword.



Means that base use case includes inclusion use case
Or base use case has inclusion use case as a part of it

Why do we use <<include>>?

- Using `<<include>>` removes the need for tedious cut-and-paste operations between use case descriptions, since updates are made in only one place instead of every use case.
- The `<<include>>` relationship gives you a good indication at system design time that the implementation of `Check Identity` will need to be a reusable part of your system.

WORKSHEET–Exercise 1

Use Case Generalization

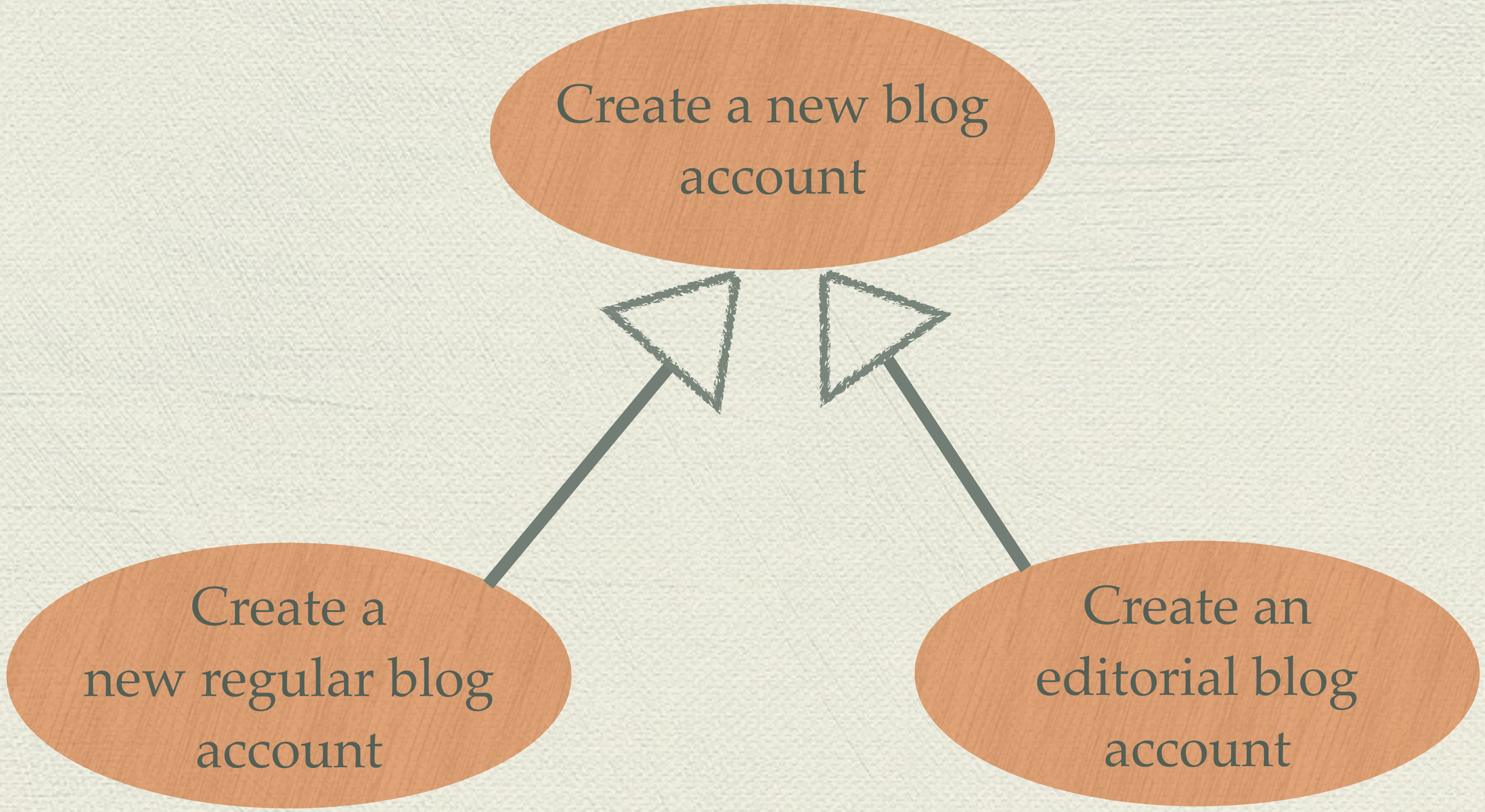
- Sometimes you'll come across a use case whose behavior can be applied to several different cases, but with small changes.
- For example, if the CMS supports several different types of blog accounts, such as a regular account with one blog or an editorial account that can make changes to entries in a set of blogs.

Create a new regular blog account

1. The Administrator asks the system to create a new blog account.
2. **The Administrator selects a regular account type.**
3. The Administrator enters the author's details.
4. The author's details are verified using the Author Credentials Database.
5. The new blog account is created.
6. A summary of the new blog account's details are emailed to the author.

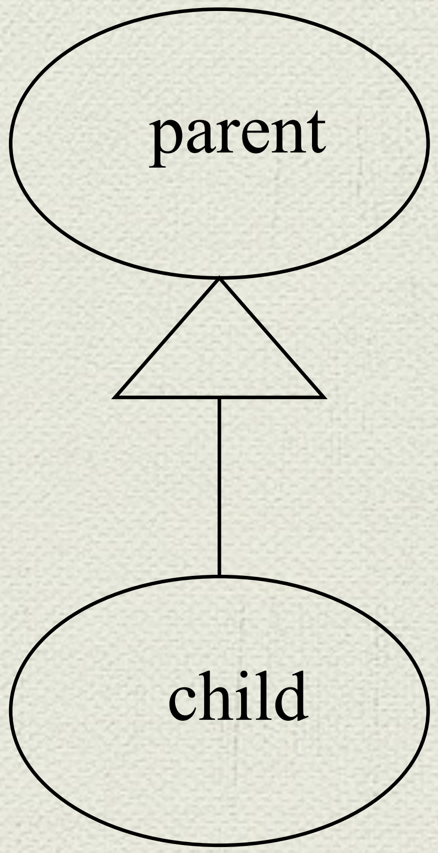
Create an editorial blog account

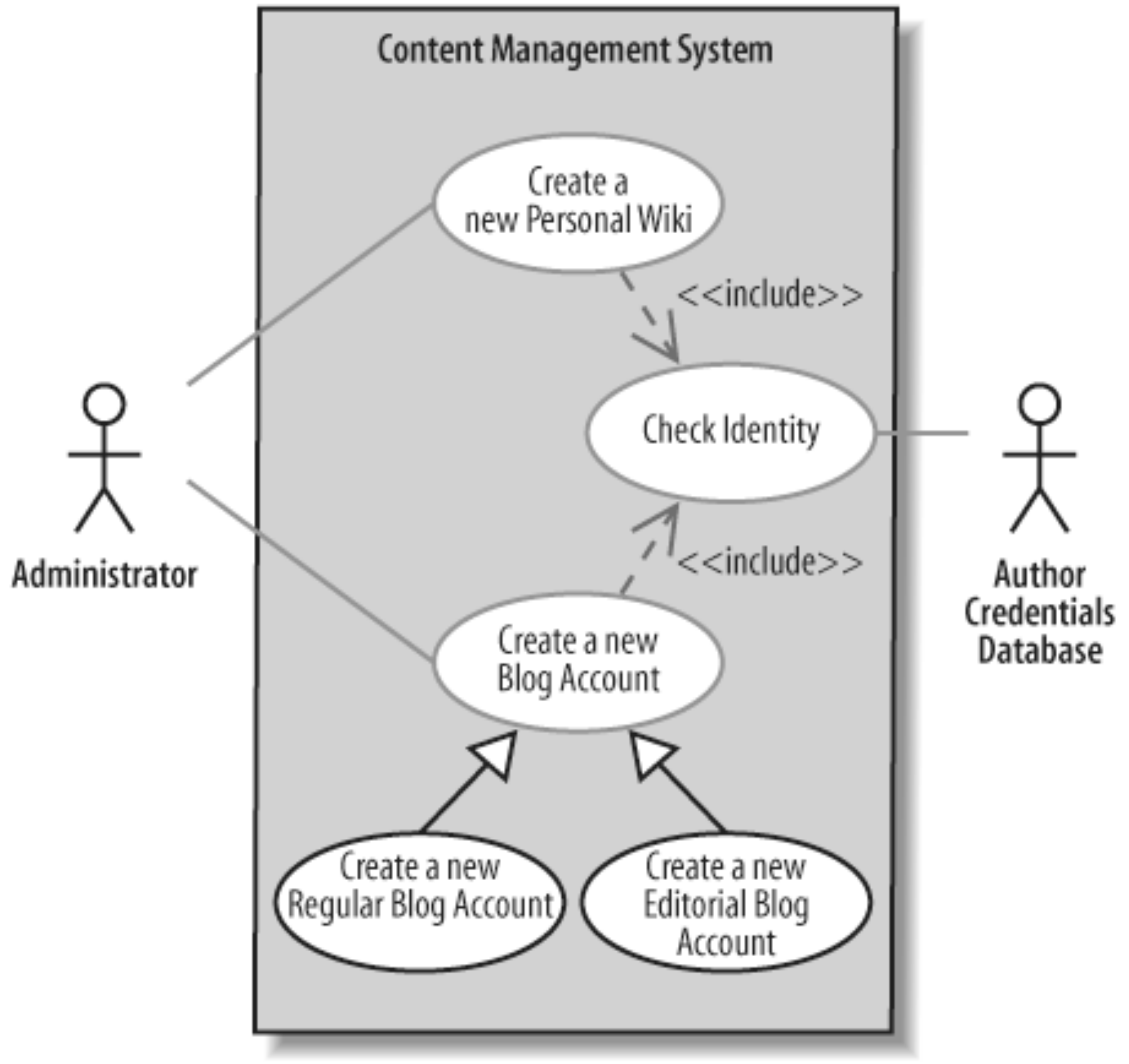
1. The Administrator asks the system to create a new blog account.
2. **The Administrator selects an editorial account type.**
3. The Administrator enters the author's details.
4. **The Administrator selects the blogs that the account is to have editorial rights over.**
5. The author's details are verified using the Author Credentials Database.
6. The new blog account is created.
7. A summary of the new blog account's details are emailed to the author.


Create a new blog account
Create a new regular blog account
Create an editorial blog account

Use Case Generalization (Cont.)

- You can use a use case generalization to address this situation by factoring out and reusing similar behavior from multiple use cases.
- A use case generalization is shown as a solid-line path from the more specific, or specialized use case to the more general use case, with a large triangle at the end of the path connected to the more general use case.

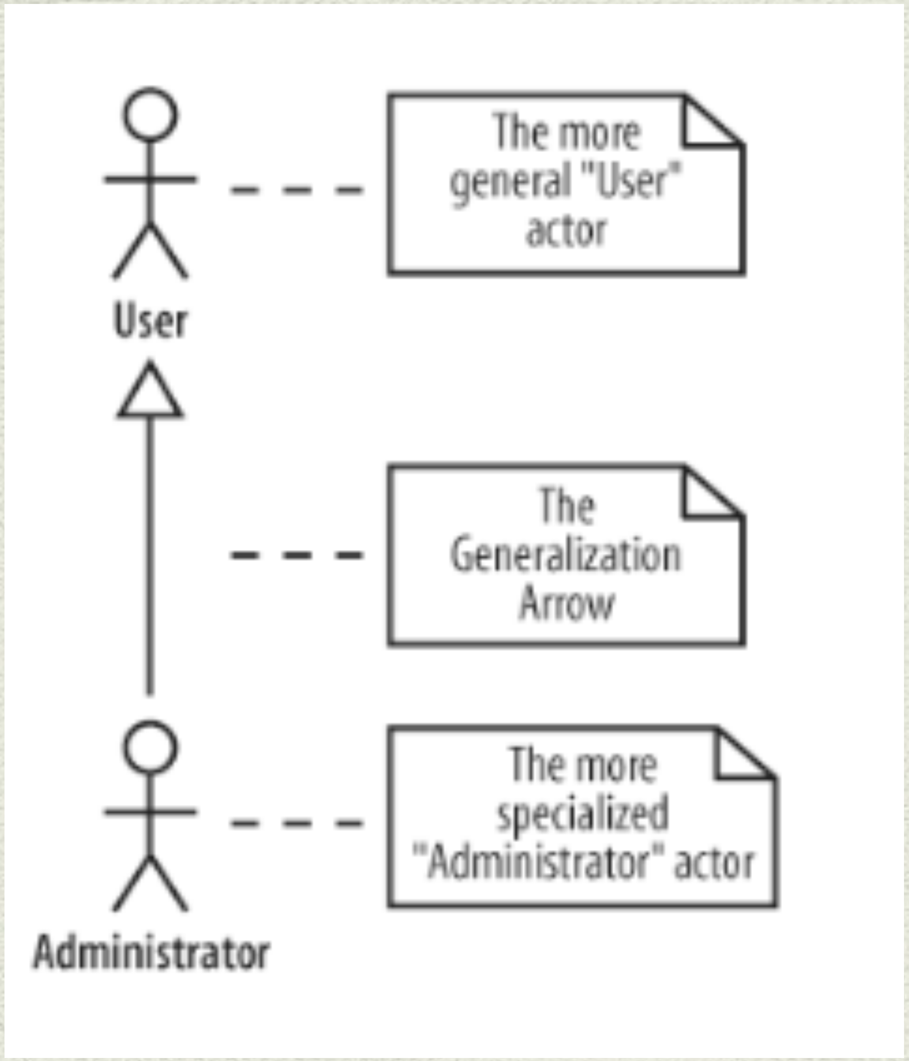


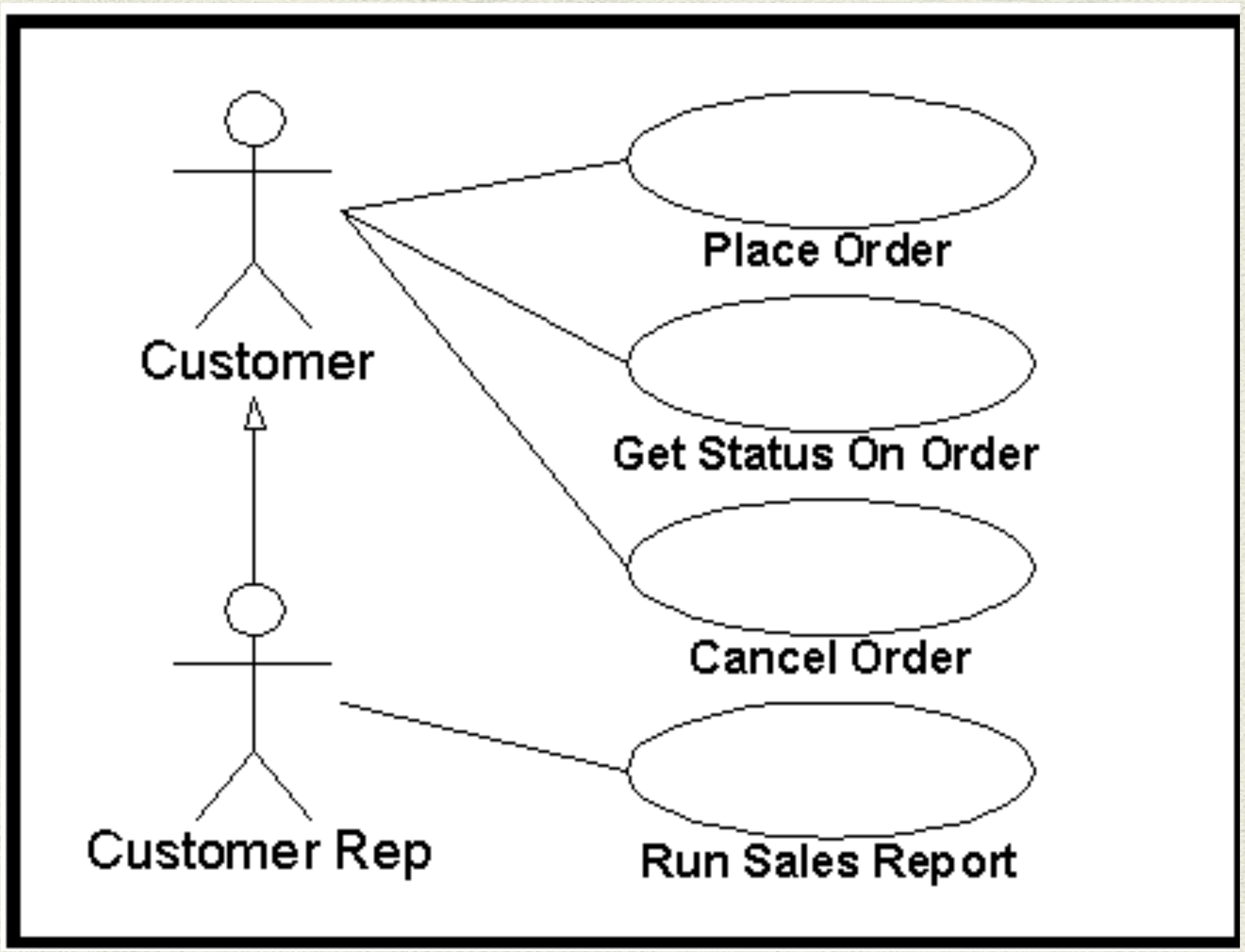

Content Management System
Create a new Personal Wiki
<<include>>
Check Identity
<<include>>
Create a new Blog Account
Create a new Regular Blog Account
Create a new Editorial Blog Account
Administrator
Author Credentials Database

WORKSHEET–Exercise 1

Generalization Between Two Actors

- Some actors are related to each other.
- For example, the `Administrator` actor is usually a special kind of system user.
- To show that an administrator can do whatever a regular user can do (with some extra additions), a generalization arrow is used.


User
The more general "User" actor
The Generalization Arrow
Administrator
The more specialized "Administrator" actor

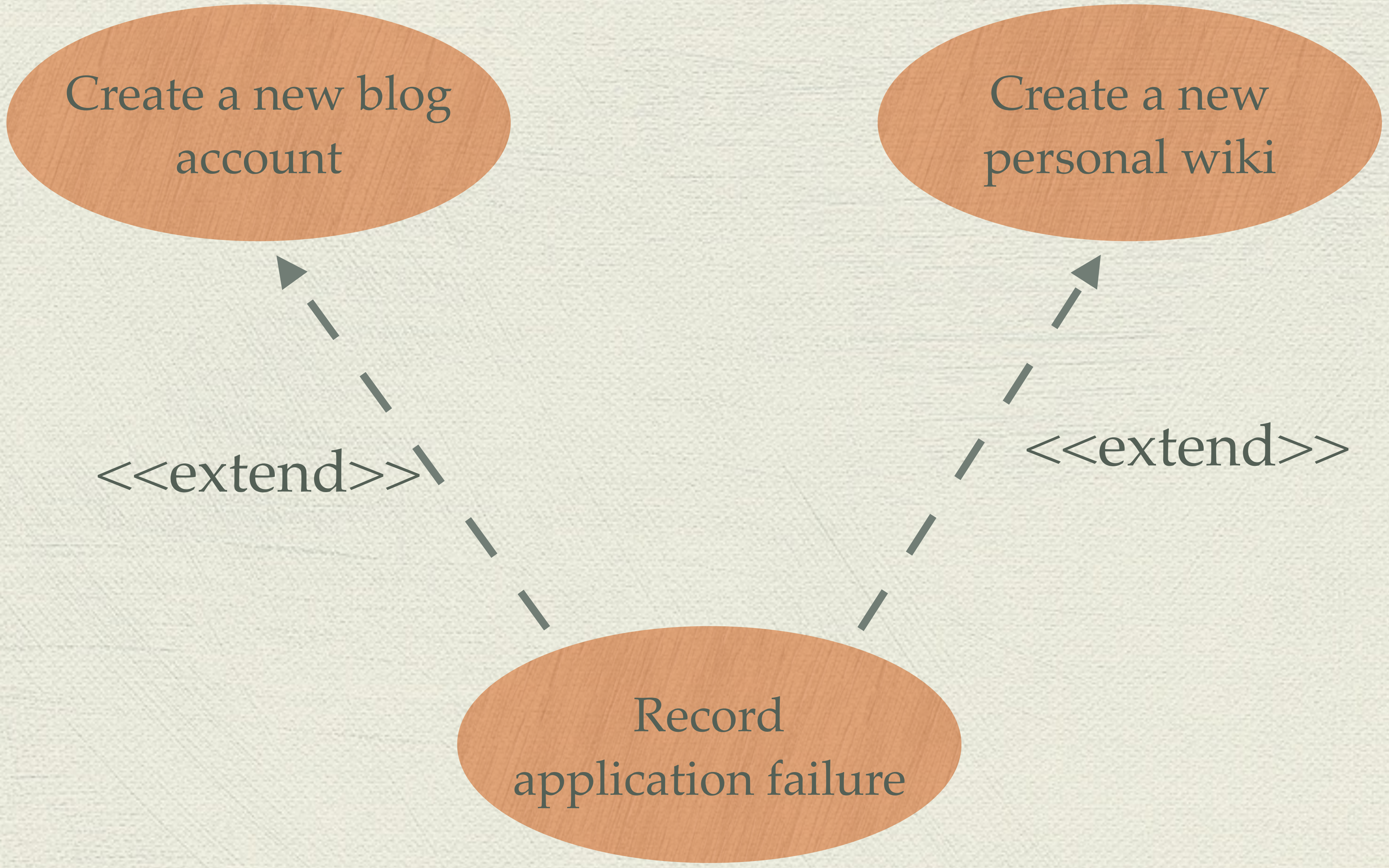


http://books.txt.com/1_3Updates/Inheritance.htm

WORKSHEET–Exercise 2

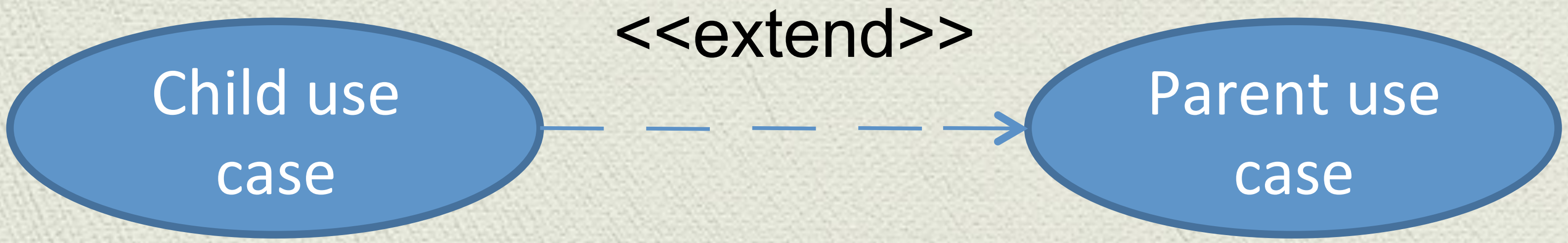
The <<extend>> Relationship

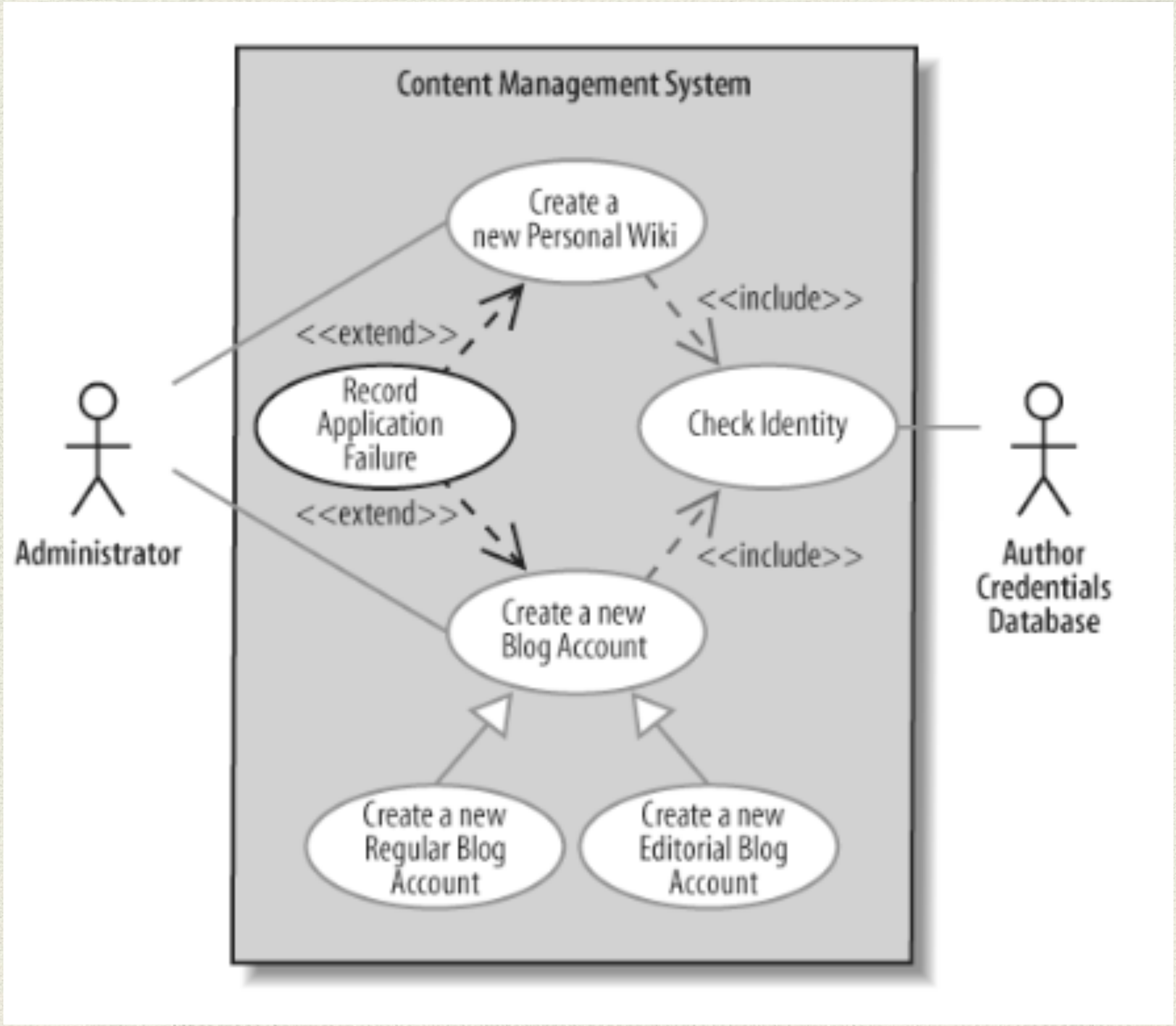
- Provides a means for you to show that a use case might completely reuse another use case's behavior, similar to the <<include>> relationship, but that this reuse was optional and dependent either on a runtime or system implementation decision
- From the CMS example, the Create a new Blog Account use case might want to record that a new author applied for an account and was rejected, adding this information to the author's application history


Create a new blog account
Create a new personal wiki
<<extend>>
<<extend>>
Record application failure

The <<extend>> Relationship (Cont.)

- In an *extend* relationship between two use cases, the child use case adds to the existing functionality and characteristics of the parent use case.
- It is depicted with a directed arrow having a dotted shaft.
- The tip of the arrowhead points to the parent use case and the child use case is connected at the base of the arrow.
- The stereotype "<<extend>>" identifies the relationship as an extend relationship.




Content Management System
Create a new Personal Wiki
<<extend>>
<<include>>
Record Application Failure
Check Identity
<<extend>>
<<include>>
Create a new Blog Account
Create a new Regular Blog Account
Create a new Editorial Blog Account
Administrator
Author Credentials Database

Use Case Diagram Relationships (Cont.)

- Relationships between use cases are more about breaking your system's behavior into manageable chunks than adding anything new to your system.
- The purpose of use case relationships is to provide your system's designers with some architectural guidance so they can efficiently break down the system's concerns into manageable pieces within the detailed system design.

References

- Alhir, S. (2003) *Learning UML*. Sebastopol: O'Reilly Media, Inc.
- Fowler, M. (2004). UML distilled: a brief guide to the standard object modeling language. Addison-Wesley Professional.
- Miles, R and Hamilton, K. (2006) Learning UML 2.0. Sebastopol: O'Reilly Media, Inc.
- UML Training Course from CRaG System, http://www.cragsystems.co.uk.