

TUM

Configuration Management Exercise

10 May 2006

Bernd Bruegge

Lilith Al-Jadiri

Technische Universität München

Institut für Informatik

Lehrstuhl für Angewandte Softwaretechnik

<http://www.bruegge.in.tum.de/static/contribute/Lehrstuhl/SE2SoSe2006.htm>

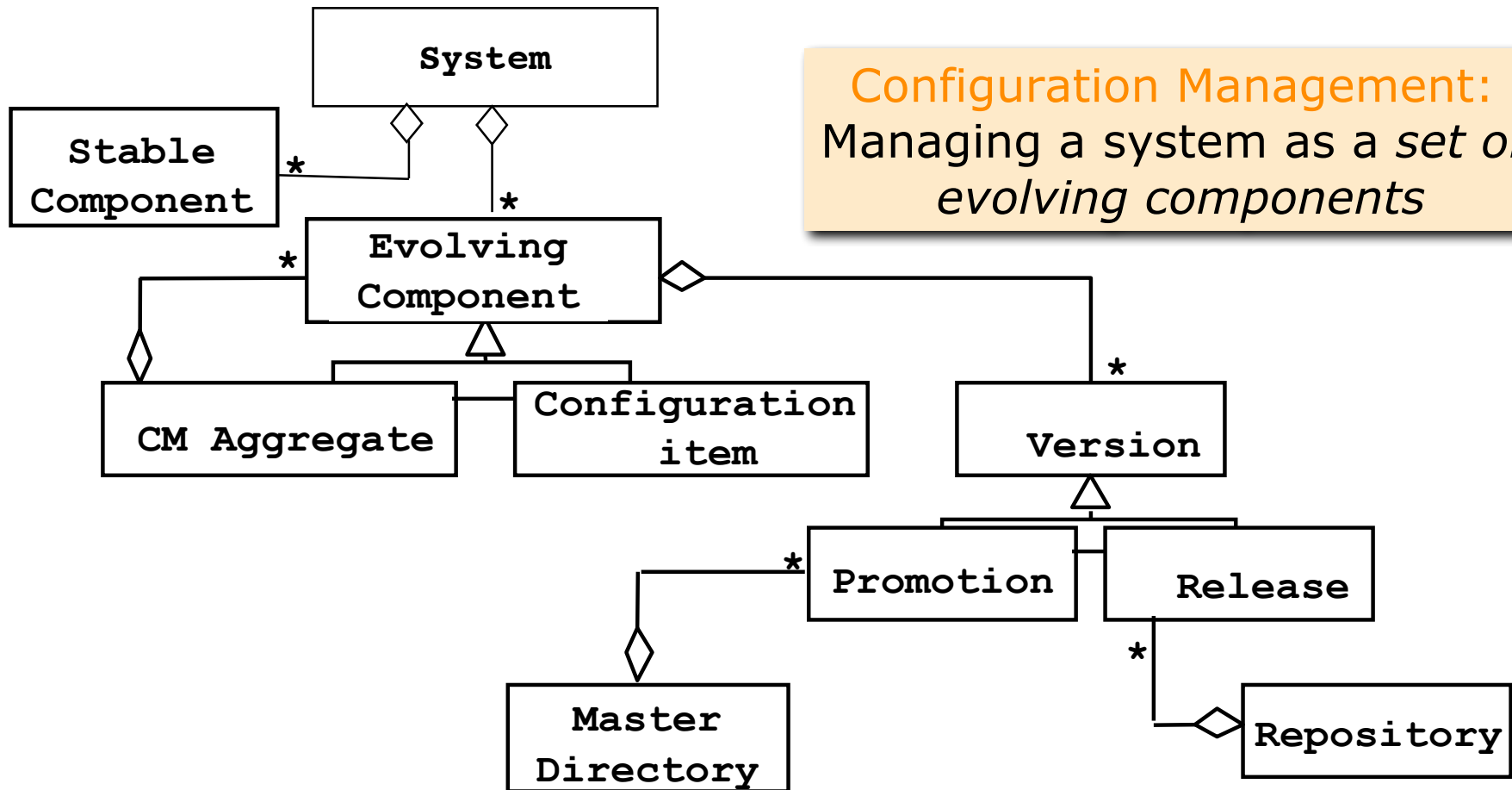
Outline of the Lecture

- ✓ Purpose of Software Configuration Management (SCM)
- ✓ Some Terminology
- ✓ Software Configuration Management Activities
 - Outline of a Software Configuration Management Plan
 - Configuration Management Tools (previously handled in SE1)

Odds and Ends

- *Submissions of homeworks in teams of 2*
- Naming of homework submissions
 - *Subject:*
HW#_FirstName_LastName_FirstName_LastName
 - *Example:*
Subject HW#2_Lilith_AlJadiri_Bernd_Bruegge
- *Acceptable Formats:*
 - *Only PDF-Format!*
 - *Or Hardcopy*
- *Submission Address:*
 - *aljadiri@in.tum.de*

Configuration Management Model: Revision



Configuration Management:
Managing a system as a set of
evolving components

(Slide 35 in the lecture)

Software Configuration Management Planning

- Software configuration management planning starts during the early phases of a project
- The outcome of the SCM planning phase is the *Software Configuration Management Plan (SCMP)* which might be extended or revised during the rest of the project
- The SCMP can either follow a public standard like the IEEE 828, or an internal (e.g. company specific) standard.

The Software Configuration Management Plan

- Defines the *types of evolving components* to be managed and a document naming scheme
- Defines *who takes responsibility* for the configuration management procedures and creation of baselines
- Defines *policies for change* control and version management
- Describes the *tools* which should be used to assist the configuration management process and any limitations on their use
- Defines the *configuration management database* used to record configuration information.

Outline of a Software Configuration Management Plan (SCMP, IEEE 828-1990)

1. Introduction

- Describes purpose, scope of application, key terms and references

2. Management (WHO?)

- Identifies the responsibilities and authorities for accomplishing the planned configuration management activities

3. Activities (WHAT?)

- Identifies the activities to be performed in applying to the project

4. Schedule (WHEN?)

- Establishes the sequence and coordination of the SCM activities with project mile stones

5. Resources (HOW?)

- Identifies tools and techniques required for the implementation of the SCMP

6. Maintenance

- Identifies activities and responsibilities on how the SCMP will be kept current during the life-cycle of the project.

SCMP Section 1: Introduction

- Overview of configuration management activities
- Scope
- Identification of other software to be included as part of the SCMP
- Relationship of software configuration management to hardware configuration management activities
- Degree of formality and depth of control
- Limitations and constraints for applying configuration management to this project
- Assumptions that impact on cost and schedule.



SCMP Section 2: Management

2.1 Organization

- Organizational context (technical and managerial) within which the configuration management activities are implemented

2.2. Responsibilities

- List name or job title of people how perform activities
- For each board, list
 - purpose and objectives
 - membership and affiliations
 - scope of authority
 - operational procedures

3. Applicable Policies:

- External constraints placed on the SCMP.

SCMP Section 3: Configuration Management Activities

- 3.1 Configuration Identification (see lecture slides)
- 3.2 Configuration Control
- 3.3 Configuration Status Accounting
- 3.4 Configuration Audits and Reviews
- 3.5 Interface Control (not treated in this lecture)

3.2 Configuration Control

- Defines the following steps
 - 3.2.1 How to identify the need for a change (layout of change request form)
 - 3.2.2 Analysis and evaluation of a change request
 - 3.2.3 Approval or disapproval of a request
 - 3.2.4 Verification, implementation and release of a change

3.2.1 Change Request

- Specifies the procedures for requesting a change to a base-lined configuration items and the information to be documented:
 - Name(s) and version(s) of the configuration item(s) where the problem appears
 - Originator's name and address
 - Date of request
 - Indication of urgency
 - The need for the change
 - Description of the requested change

3.2.2 Evaluation of a Change

- Specifies the analysis required to determine the impact of proposed changes and the procedure for reviewing the results of the analysis.

3.2.3 Change Approval or Disapproval

- Describes the organization of the configuration control board (CCB).
- Configuration Control Board (CCB)
 - Can be an individual or a group.
 - Multiple levels of CCBs are also possible, depending on the complexity of the project
- Multiple levels of CCBs may be specified.
 - In small development efforts one CCB level is sufficient.
- Also indicates the level of authority of the CCB and its responsibility.
 - In particular, it must be specified when the CCB is invoked.

3.2.4 Implementing Change

- Specifies the activities for verifying and implementing an approved change.
- A completed change request must contain this information:
 - The original change request(s)
 - The names and versions of the affected configuration items
 - Verification date and responsible party
 - Identifier of the new version
 - Release or installation date and responsible party
- This section must also specify activities for
 - Archiving completed change requests
 - Planning and control of releases
 - How to coordinate multiple changes
 - How to add new configuration items to the configuration
 - How to deliver a new baseline

3.3 Configuration Status Accounting

- Answers the following questions:
 - What elements are to be tracked and reported for baselines and changes?
 - What types of status accounting reports are to be generated? What is their frequency?
 - How is information to be collected, stored and reported?
 - How is access to the configuration management status data controlled?

3.4 Configuration Audits and Reviews

- Identifies audits and reviews for the project.
 - An audit determines for each configuration item if it has the required physical and functional characteristics.
 - A review is a management tool for establishing a baseline.
- For each audit or review the plan has to define:
 - Objective
 - The Configuration Items under review
 - The schedule for the review
 - Procedures for conducting the review
 - Participants by job title
 - Required documentation
 - Procedure for recording deficiencies and how to correct them
 - Approval criteria

Tasks for Configuration Managers

Write the SCMP

Define configuration items

Define promote /release policies

Define activities

Tailoring the SCMP

- The IEEE standard allows quite a bit flexibility for preparing an SCMP
- The SCMP may be
 - tailored upward:
 - to add information
 - to use a specific format
 - tailored downward
 - Some SCMP components might not apply to a particular project.
 - “The incredibly shrinking SCMP”

Conformance to IEEE Standard 828-1990

- Presentation format & Minimum information
 - A separate document or a section embedded in another document titled "Software Configuration Management Plan"
- Consistency Criteria:
 - All defined SCM activities (Section 3.1 to 3.6) are assigned to a participant and resources to accomplish the activities
 - All identified configuration items have defined processes for baseline establishment and change control (Section 3.2)
- If the above criteria are met, we can write:
"This SCMP conforms with the requirements of IEEE Std 828-1990."

Tasks for Configuration Managers (Summary)

SCMP following the IEEE 828-1990 standard

Define configuration items

Define promote /release policies

Define activities and responsibilities

Set up configuration management system

References

- Readings used for this lecture
 - [IEEE Std 828] Software Configuration Management
 - [IEEE Std 1042] Guide to Configuration Management Plan (SCMP)
- Additional References
 - CVS **<http://www.cvshome.org/>**
 - Online Documentation:
<http://www.cvshome.org/docs/manual/cvs.html>
 - Subversion: <http://subversion.tigris.org/>

Exercise

- Write a Software Configuration Management Plan (SCMP) that conforms with the requirements of IEEE Std 828-1990 for a project of your choice
- Due Date: 16 May 2006

Summary

- UML Model for Configuration Management
- Software Configuration Management Plan
- Public standard for SCM plans: IEEE 828.
 - The standard can be tailored
- SCM is supported by tools. These range from
 - Simple version storage tools
 - Systems with automated procedures for policy checks and support for the creation of SCM documents.
- Examples of available tools:
 - RCS
 - CVS
 - Subversion

Additional and Backup Slides

Example SCM Plans (from IEEE 1042.1990 Guide)

Life-cycle Phase	Project Type	Size	SCM Tools	Life Span	Writing	Character of Project
A Development	Critical	Medium	Advanced	Short	Highly Structured	Complex system contracted to another company
B Concept	Prototype	Small	Basic	Short	Informal	Small software development project
C Maintenance	Support Software	Large	On-line	Full Life-Cycle	Structured	SCMP used by organization using contracted SW
D All	Commercial	Small	Integrated	Full Life-Cycle	Informal	Development of embedded applications

Software Configuration Management Tools

- RCS: very old but still in use
- CVS (Concurrent Version Control)
 - based on RCS, allows concurrent working without locking
 - <http://www.cvshome.org/>
- Perforce
 - Repository server; keeps track of developer's activities
 - <http://www.perforce.com>
- ClearCase
 - Multiple servers, process modeling, policy check mechanisms
 - <http://www.rational.com/products/clearcase/>
- Subversion
 - See next slide.

Subversion

- Open Source Project (<http://subversion.tigris.org/>)
- Based on CVS
 - Subversion interface and features similar to CVS
 - Commands: `checkout`, `add`, `delete`, `commit`, `diff`
- Differences to CVS
 - Version controlled moving, renaming and copying of files and directories
 - Version controlled metadata of files and directories
- Server Options
 - Standalone installation
 - Integrated into the Apache webserver
- The time for branch management is independent of the size of the system (unlike CVS, which creates physical copies of the files, Subversion uses only tags)

Download Areas

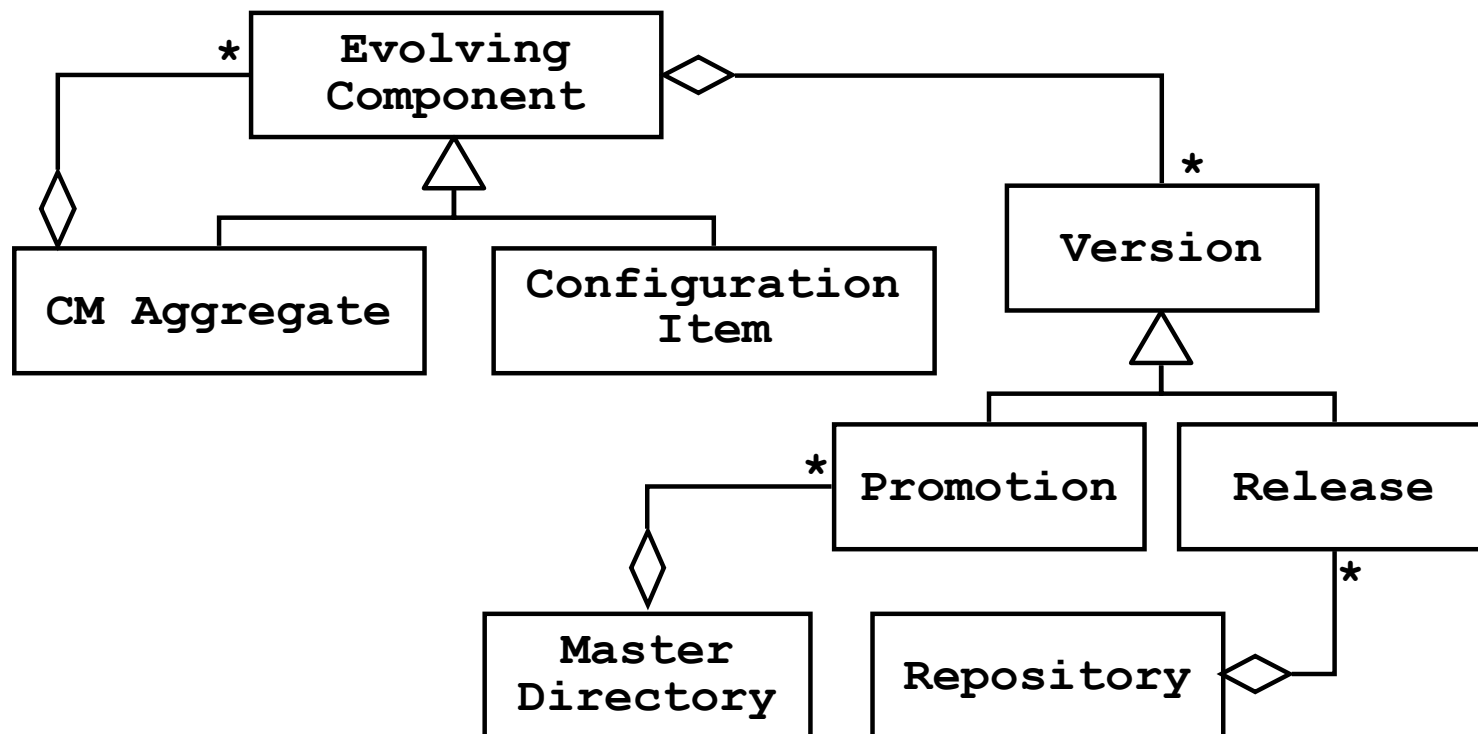
- Windows Client:  **TortoiseSVN**
 - Open Source Project
 - Graphical client that integrates into the Windows file explorer
 - <http://tortoisesvn.tigris.org>
- Unix command line clients:  Red Hat Linux and Fedora Core  SuSE SUSE Linux  SOLARIS
 FreeBSD
 - http://subversion.tigris.org/project_packages.html
- Mac OS X Client:
 - Command line client with a graphical installer
 - <http://www.codingmonkeys.de/mbo/>

Configuration management tools

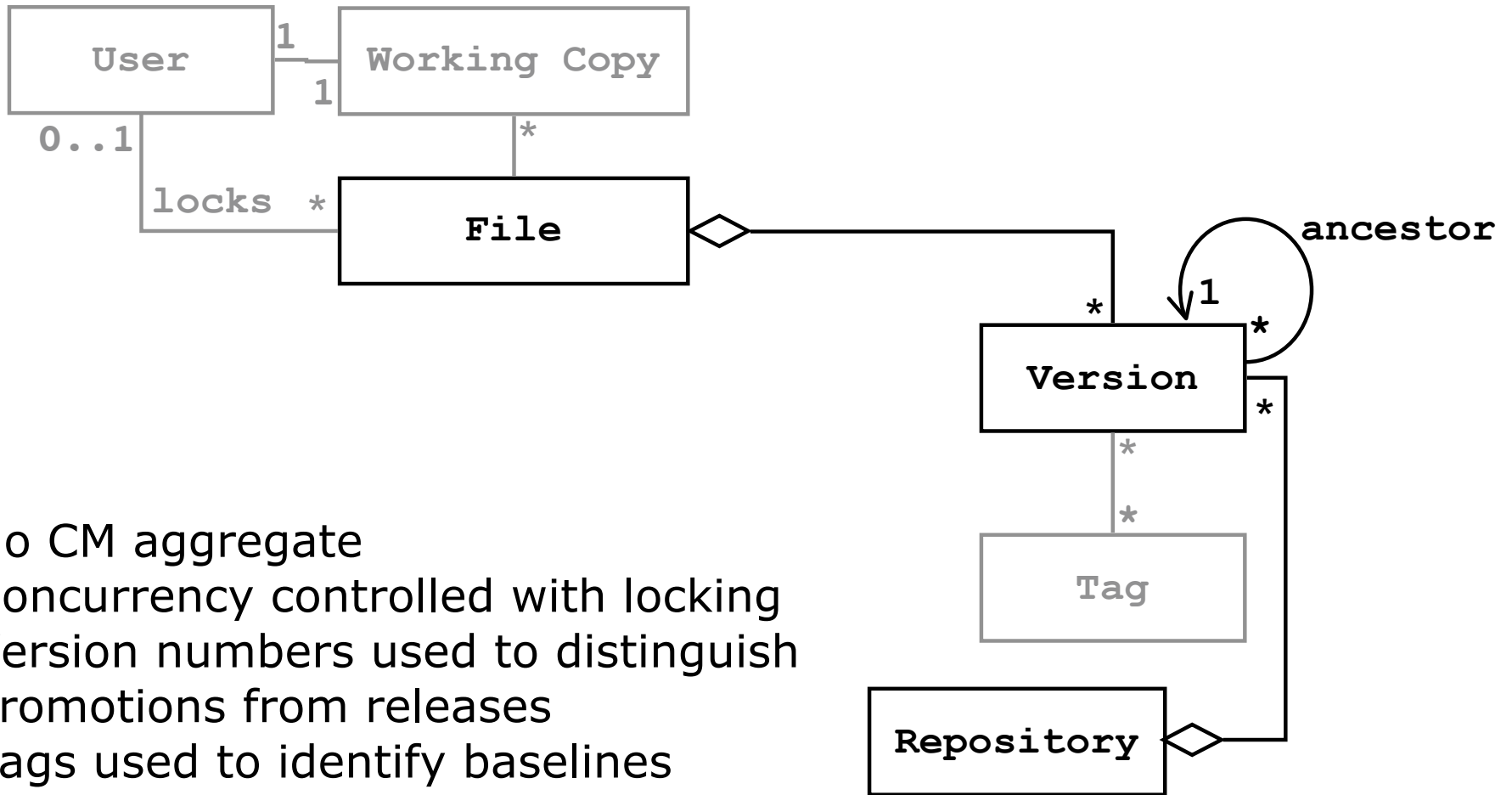
Overview

- ▶ IEEE Configuration Management Model
- ▶ Revision Control System (RCS)
- ▶ Concurrent Version System (CVS)
- ▶ Subversion (SVN)

Full Model for Configuration (UML Class Diagram)

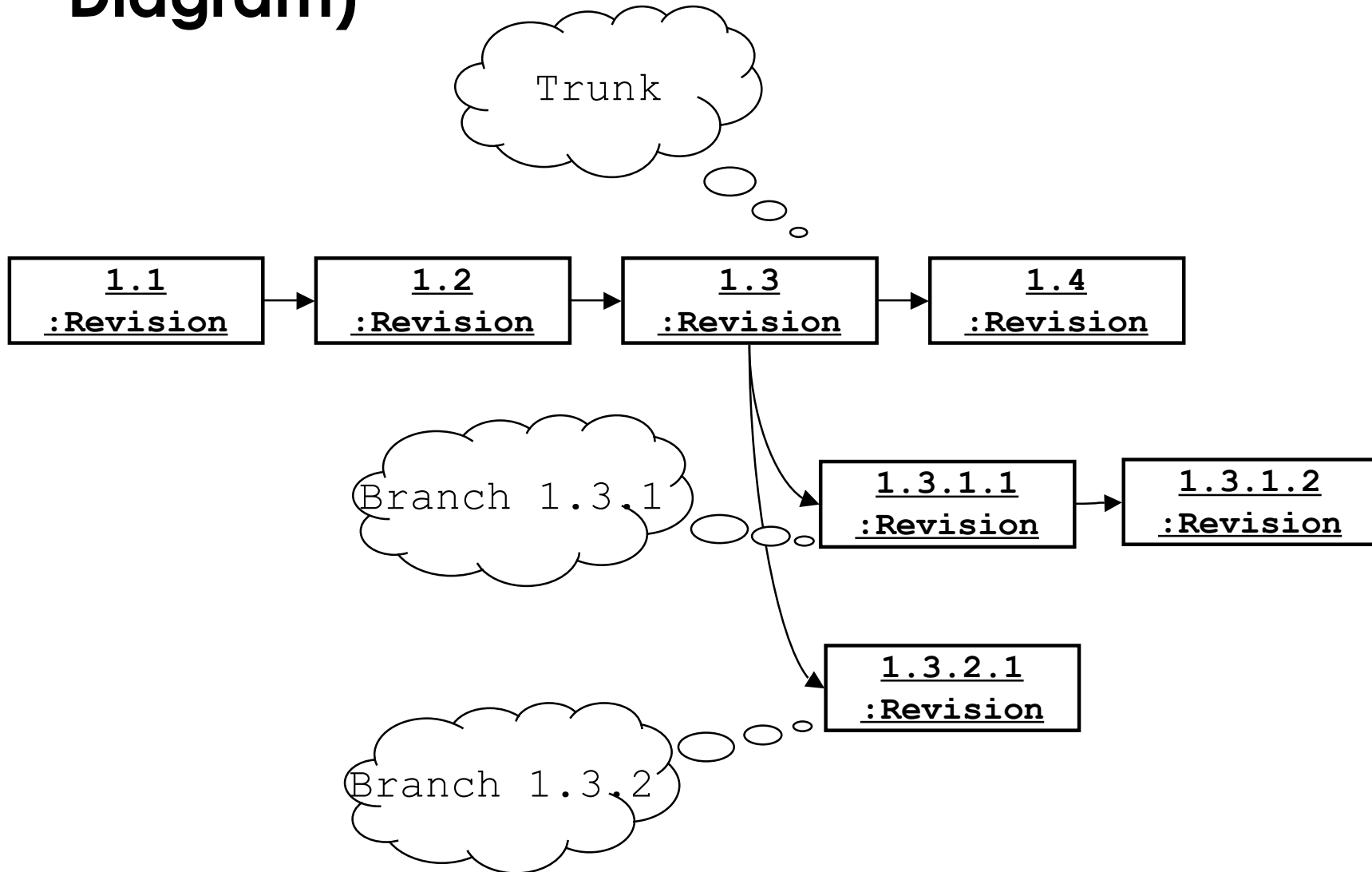


RCS Concepts (UML Class Diagram)

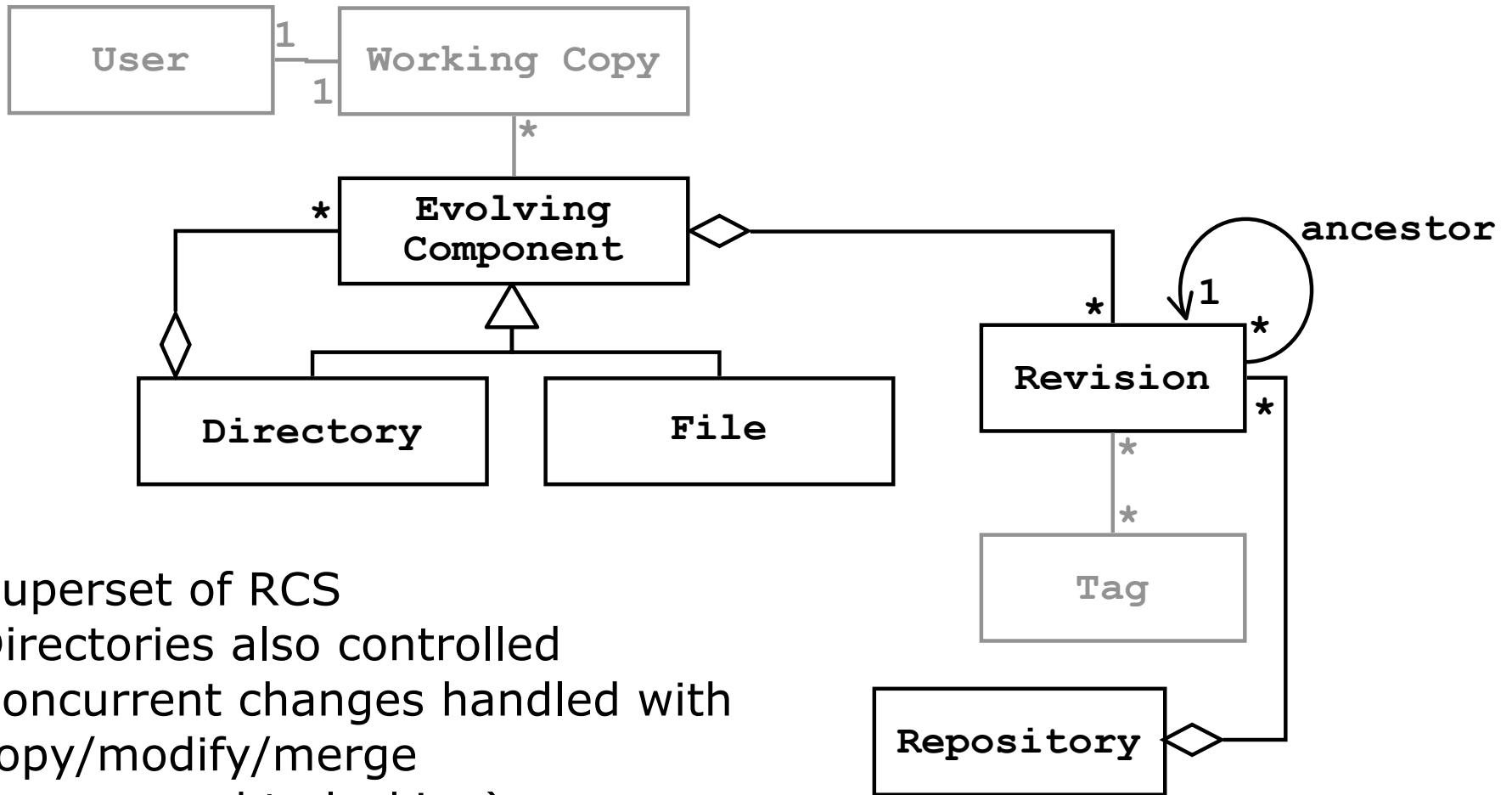


No CM aggregate
Concurrency controlled with locking
Version numbers used to distinguish promotions from releases
Tags used to identify baselines

RCS Version Numbering (UML Instance Diagram)

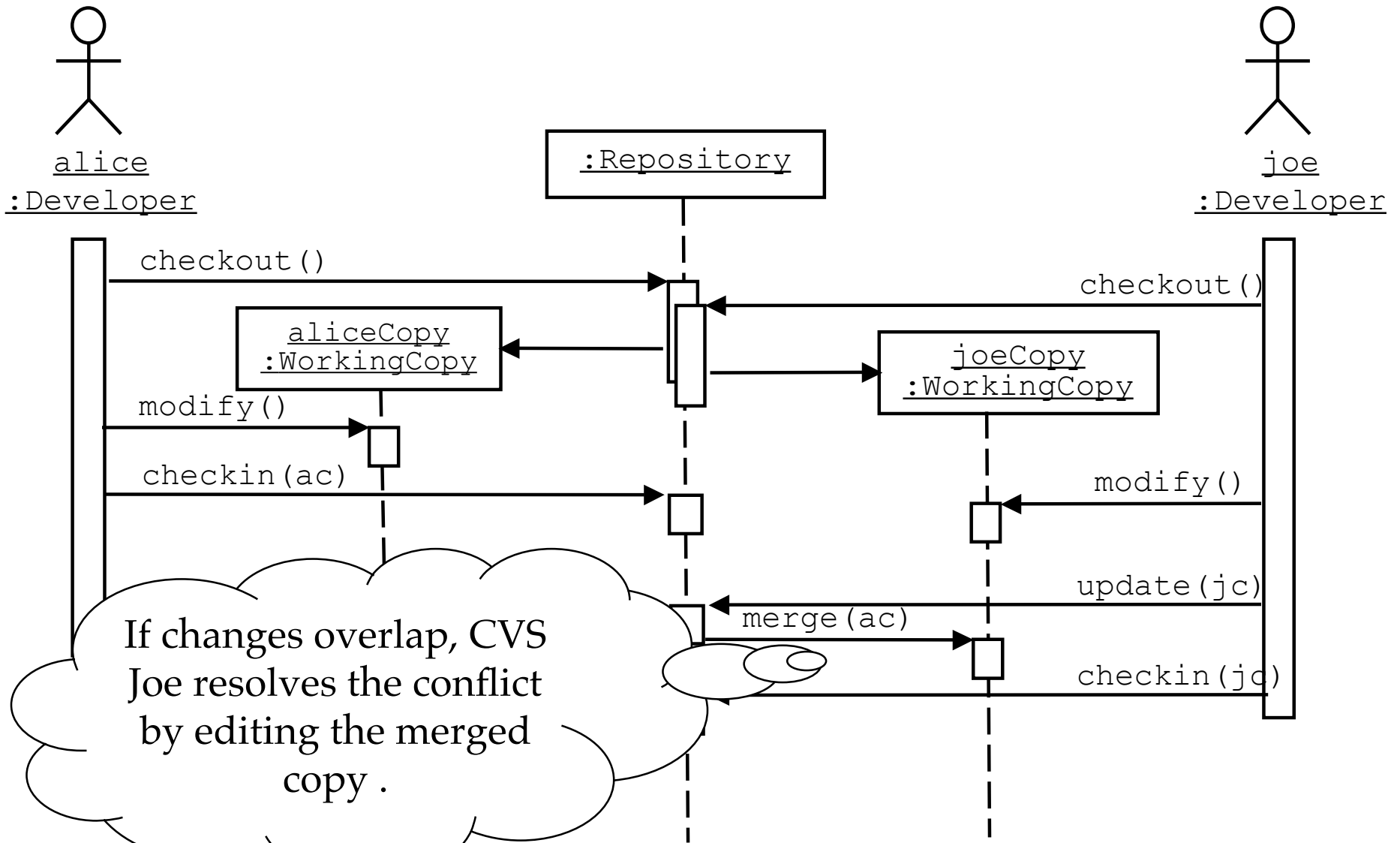


CVS Concepts (UML Class Diagram)

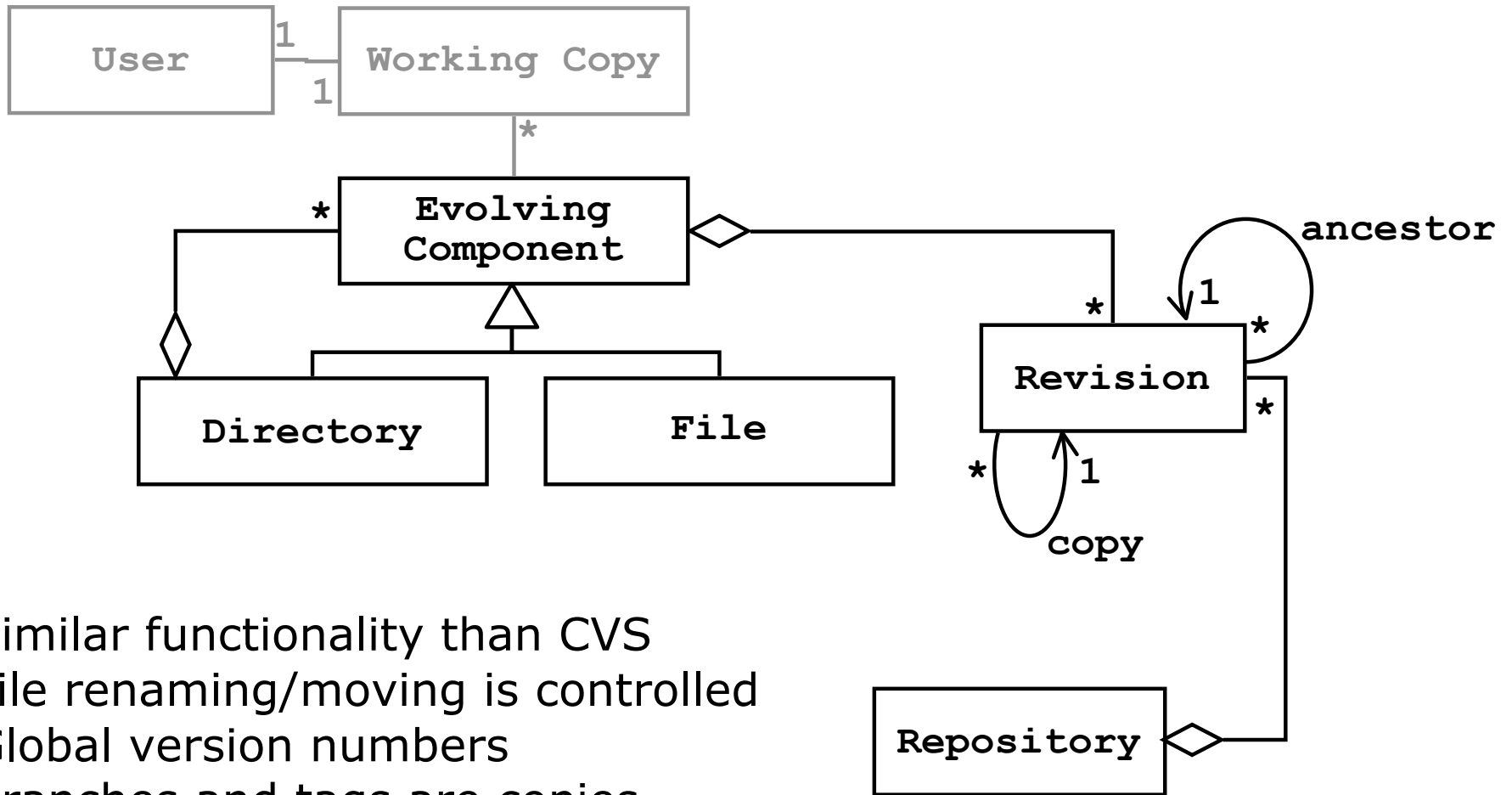


Superset of RCS
 Directories also controlled
 Concurrent changes handled with
 copy/modify/merge
 (as opposed to locking)

CVS Copy/Modify/Merge

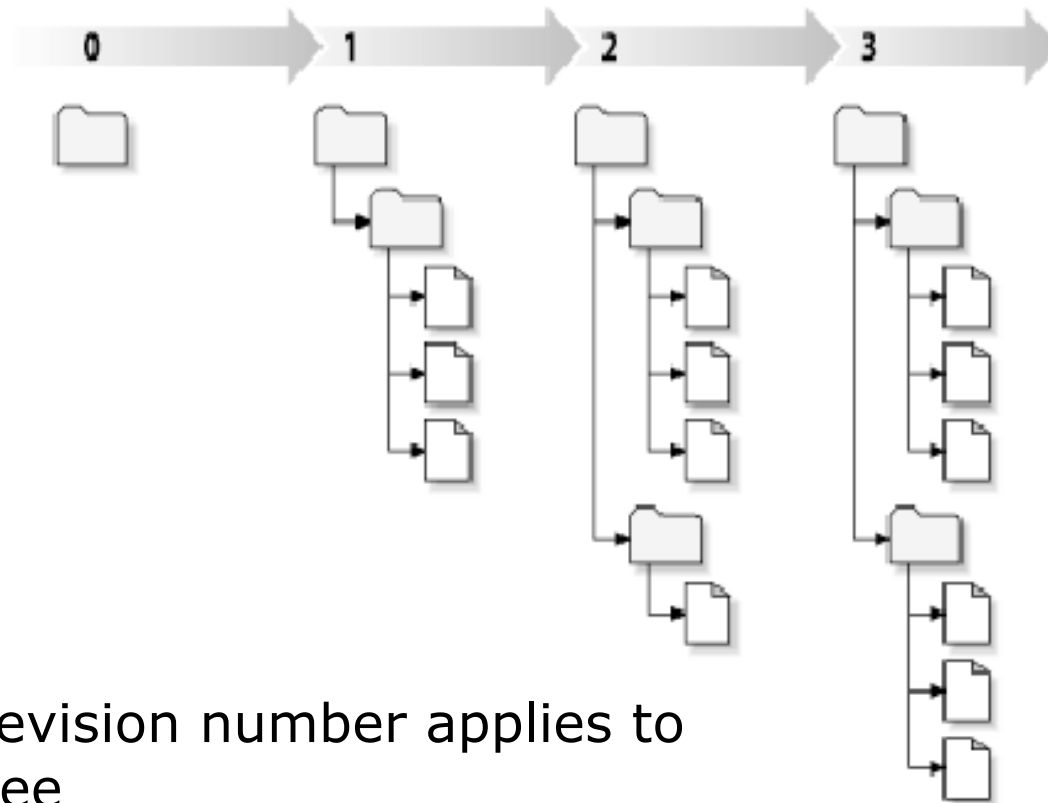


SVN Concepts (UML Class Diagram)



Similar functionality than CVS
 File renaming/moving is controlled
 Global version numbers
 Branches and tags are copies

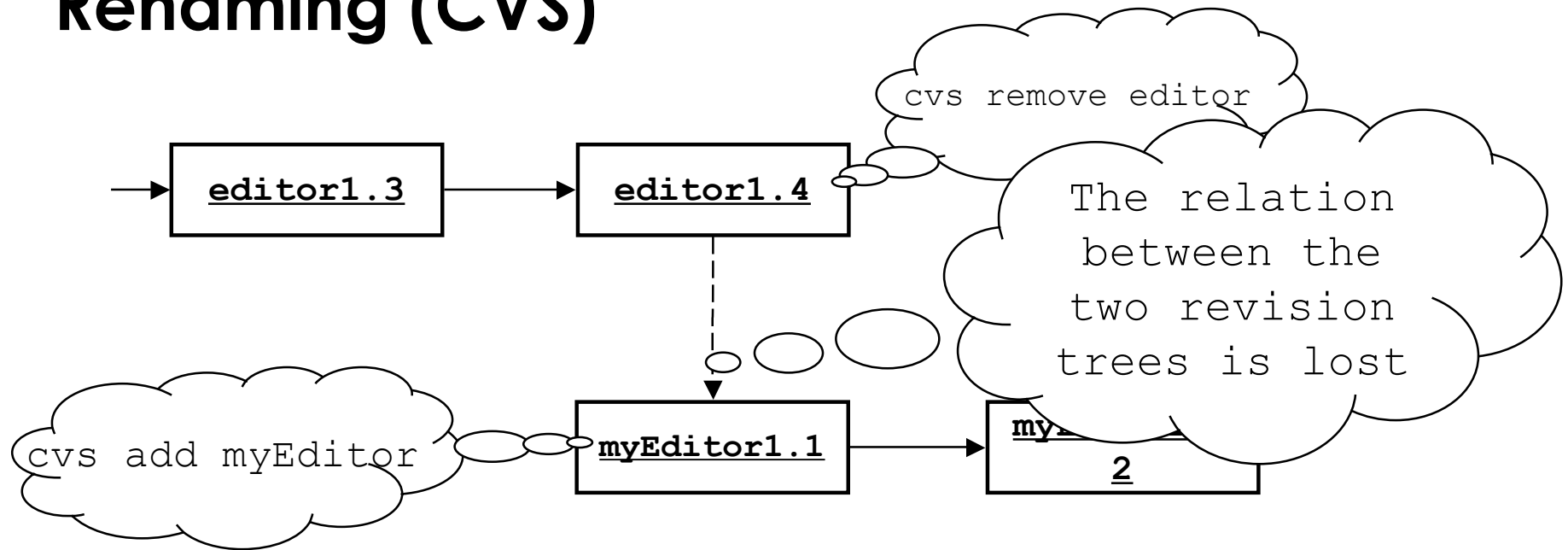
SVN Global Revision Numbering



The global revision number applies to the entire tree

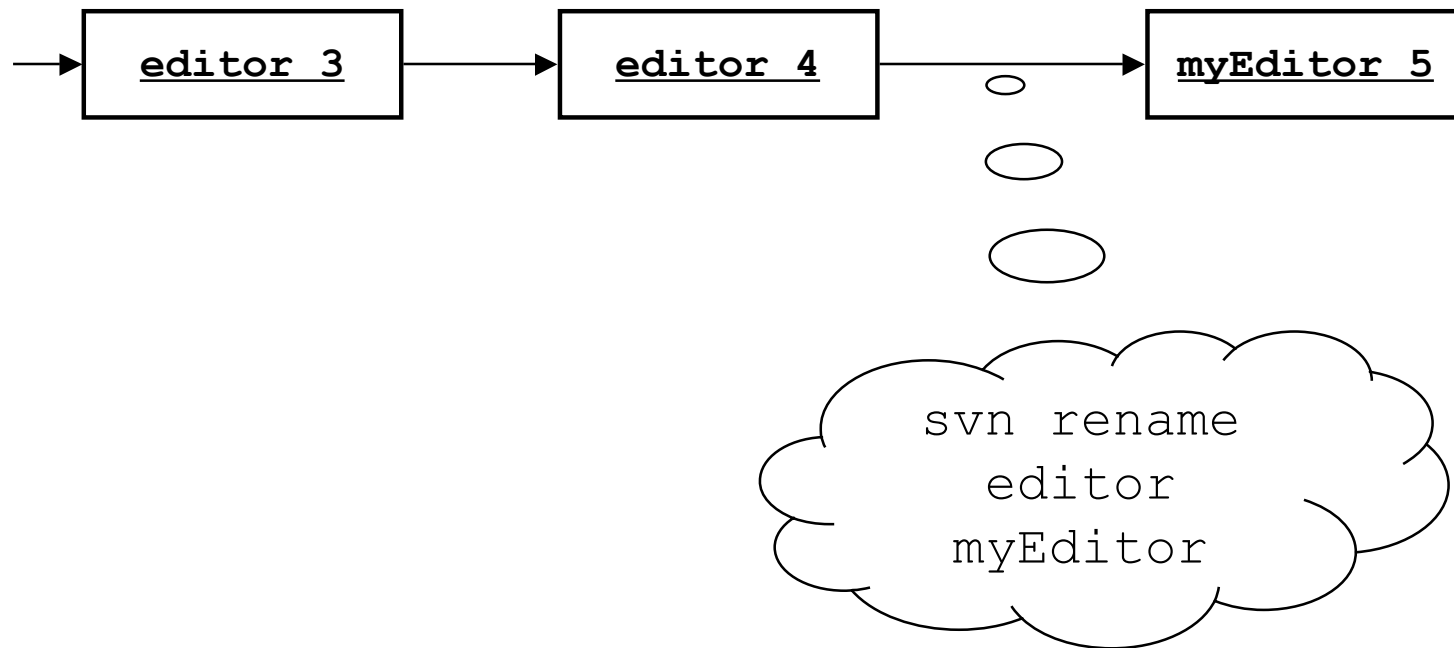
The revision number is increased for each check in, regardless of the extent of the changes.

Renaming (CVS)



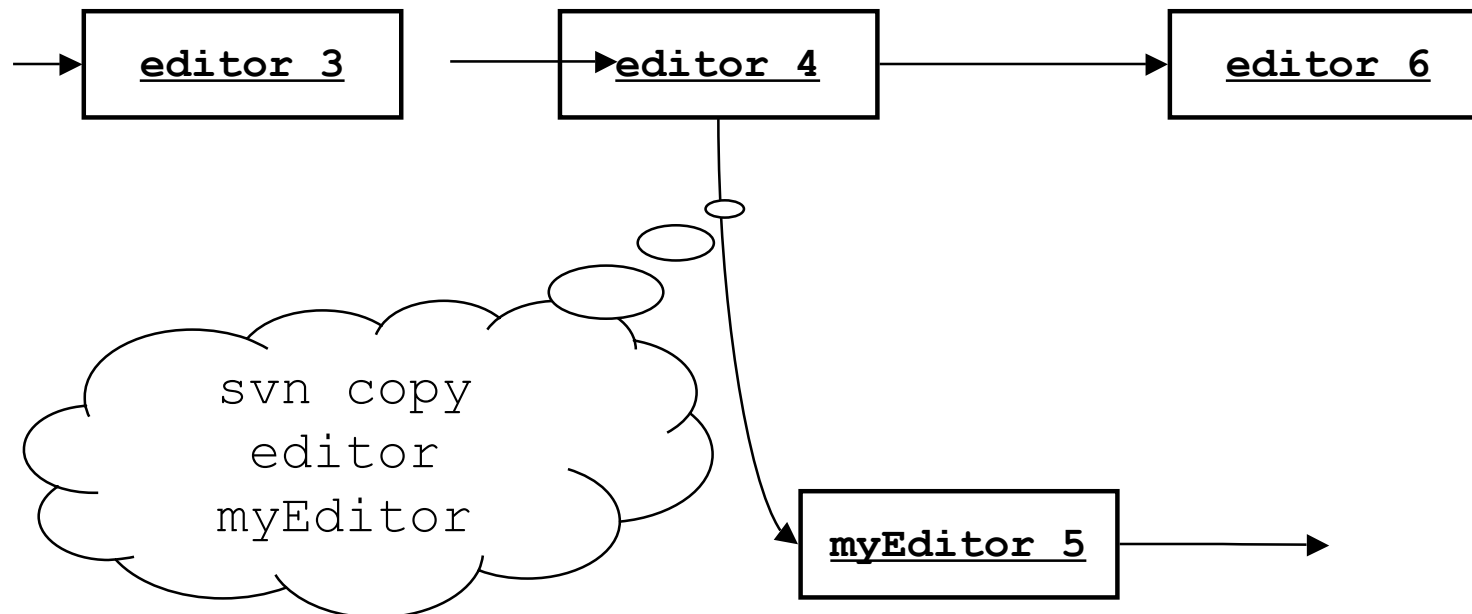
- CVS does not support renaming
- Instead, the end user
 - renames the file in the working copy
 - removes the old name from the repository
 - adds the new name in the repository
- The two history trees are disconnected.

Renaming (Subversion)



- Subversion treats the name of a configuration item as an attribute
- Name changes are dealt the same as a content change
- There is a single history tree
- Renaming conflicts (two users renaming the same file at the same time) are detected.

Copying (Subversion)



- Subversion tracks copies in a similar way than renaming.
- The history of a configuration item includes the history of the original it was copied from.
- Copies are cheap, differences (i.e., the name change) are stored.
- Branches in subversion are simple copies.