Software Configuration Management

Addendum zu Kapitel 13

University of Zurich
Department of Informatics

s.e.a.l.
software evolution & architecture lab

Outline

- **Purpose of Software Configuration Management (SCM)**
  - Motivation: Why software configuration management?
  - Definition: What is software configuration management?
  - Activities and roles in software configuration management
- **Some Terminology**
  - Configuration Item, Baseline, SCM Directory, Version, Revision Release.
- **Software Configuration Management Activities**
  - Promotion Management, Release Management, Change Management
- **Outline of a Software Configuration Management Plans**
  - Standards (Example: IEEE 828-1990)
  - Basic elements of IEEE 828-1990
- **Configuration Management Tools**

Why Software Config Management?

- The problem:
  - Multiple people have to work on software that is changing
  - More than one version of the software has to be supported:
    - Released systems
    - Custom configured systems (different functionality)
    - System(s) under development
  - Software must run on different machines and operating systems
- Need for coordination
- Software Configuration Management
  - manages evolving software systems
  - controls the costs involved in making changes to a system

What is Software Configuration Management?

- ### Definition:
  - A set of management disciplines within the software engineering process to develop a baseline.

  Forward Definition!

- ### Description:
  - Software Configuration Management encompasses the disciplines and techniques of initiating, evaluating and controlling change to software products during and after the software engineering process.

- ### Standards (approved by ANSI)
  - IEEE 828: Software Configuration Management Plans
  - IEEE 1042: Guide to Software Configuration Management

University of Zurich
Department of Informatics

Software Configuration Management is a Project Function

- SCM is a Project Function (as defined in the SPMP) with the goal to make technical and managerial activities more effective.
- Software Configuration Management can be administered in several ways:
  - A single software configuration management team for the whole organization
  - A separate configuration management team for each project
  - Software Configuration Management distributed among the project members
  - Mixture of all of the above

© 2004 B. Brügge, A. Dutoit, Pearson Education

Configuration Management Activities

- Software Configuration Management Activities:
  - Configuration item identification
  - Promotion management
  - Release management
  - Branch management
  - Variant management
  - Change management

- No fixed rules:
  - Activities are usually performed in different ways (formally, informally) depending on the project type and life-cycle phase (research, development, maintenance).

Configuration Management Activities

- Configuration item identification
  - modeling of the system as a set of evolving components
- Promotion management
  - is the creation of versions for other developers
- Release management
  - is the creation of versions for the clients and users
- Change management
  - is the handling, approval and tracking of change requests
- Branch management
  - is the management of concurrent development
- Variant management
  - is the management of versions intended to coexist

© 2004 B. Brügge, A. Dutoit, Pearson Education

Configuration Management Roles

- ## Configuration Manager
  - Responsible for identifying configuration items. The configuration manager can also be responsible for defining the procedures for creating promotions and releases

- ## Change control board member
  - Responsible for approving or rejecting change requests

- ## Developer
  - Creates promotions triggered by change requests or the normal activities of development. The developer checks in changes and resolves conflicts

- ## Auditor
  - Responsible for the selection and evaluation of promotions for release and for ensuring the consistency and completeness of this release

University of Zurich
Department of Informatics

Terminology

- We will define the following terms
  - Configuration Item
  - Baseline
  - SCM Directories
  - Version
  - Revision
  - Release

⇨ The definition of the terms follows the IEEE standard.

⇨ Different configuration management systems may use different terms.

  ⇨ Example: CVS configuration management system used in our projects uses terms differing from the IEEE standard.

Terminology: Configuration Item

"An aggregation of hardware, software, or both, that is designated for configuration management and treated as a single entity in the configuration management process."

❖ Software configuration items are not only program code segments but all type of documents according to development, e.g

  ➭ all type of code files
  ➭ drivers for tests
  ➭ analysis or design documents
  ➭ user or developer manuals
  ➭ system configurations (e.g. version of compiler used)

❖ In some systems, not only software but also hardware configuration items (CPUs, bus speed frequencies) exist!

© 2004 B. Brügge, A. Dutoit, Pearson Education

Tasks for the Configuration Managers

Define configuration items

Finding Configuration Items

- Large projects typically produce thousands of entities (files, documents, data ...) which must be uniquely identified.
- Any entity managed in the software engineering process can potentially be brought under configuration management control
- But not every entity needs to be under configuration management control all the time.
- Two Issues:
    - What: Selection of Configuration Items
        - What should be under configuration control?
    - When: When do you start to place entities under configuration control?
- Conflict for the Project Manager:
    - Starting with CIs too early introduces too much bureaucracy
    - Starting with CIs too late introduces chaos

University of Zurich
Department of Informatics

Finding Configuration Items (continued)

- Some items must be maintained for the lifetime of the software. This includes also the phase, when the software is no longer developed but still in use; perhaps by industrial customers who are expecting proper support for lots of years.

- An entity naming scheme should be defined
  so that related documents have related names.

- Selecting the right configuration items is a skill that takes practice
  - Very similar to object modeling
  - Use techniques similar to object modeling for finding Cis!
    - Find the CIs
    - Find relationships between CIs

University of Zurich
Department of Informatics

# Which of these Entities should be Configuration Items?

- Problem Statement
- Software Project Management Plan (SPMP)
- Requirements Analysis Document (RAD)
- System Design Document (SDD)
- Project Agreement
- Object Design Document (ODD)
- Dynamic Model
- Object model
- Functional Model
- Unit tests
- Integration test strategy

- Source code
- API Specification
- Input data and data bases
- Test plan
- Test data
- Support software (part of the product)
- Support software (not part of the product)
- User manual
- Administrator manual

Possible Selection of Configuration Items

- Problem Statement
- Software Project Management Plan (SPMP)
- ✓ Requirements Analysis Document (RAD)
- ✓ System Design Document (SDD)
- Project Agreement
- ✓ Object Design Document (ODD)
- Dynamic Model
- Object model
- Functional Model
- ✓ Unit tests
- Integration test strategy

- ✓ Source code
- API Specification
- ✓ Input data and data bases
- Test plan
- ✓ Test data
- ✓ Support software (part of the product)
- Support software (not part of the product)
- User manual
- Administrator manual

Once the Configuration Items are selected, they are usually organized in a tree

© 2004 B. Brügge, A. Dutoit, Pearson Education

Configuration Item Tree (Example)
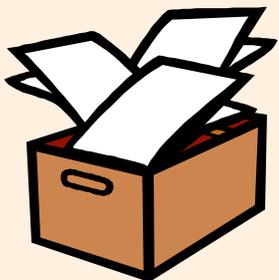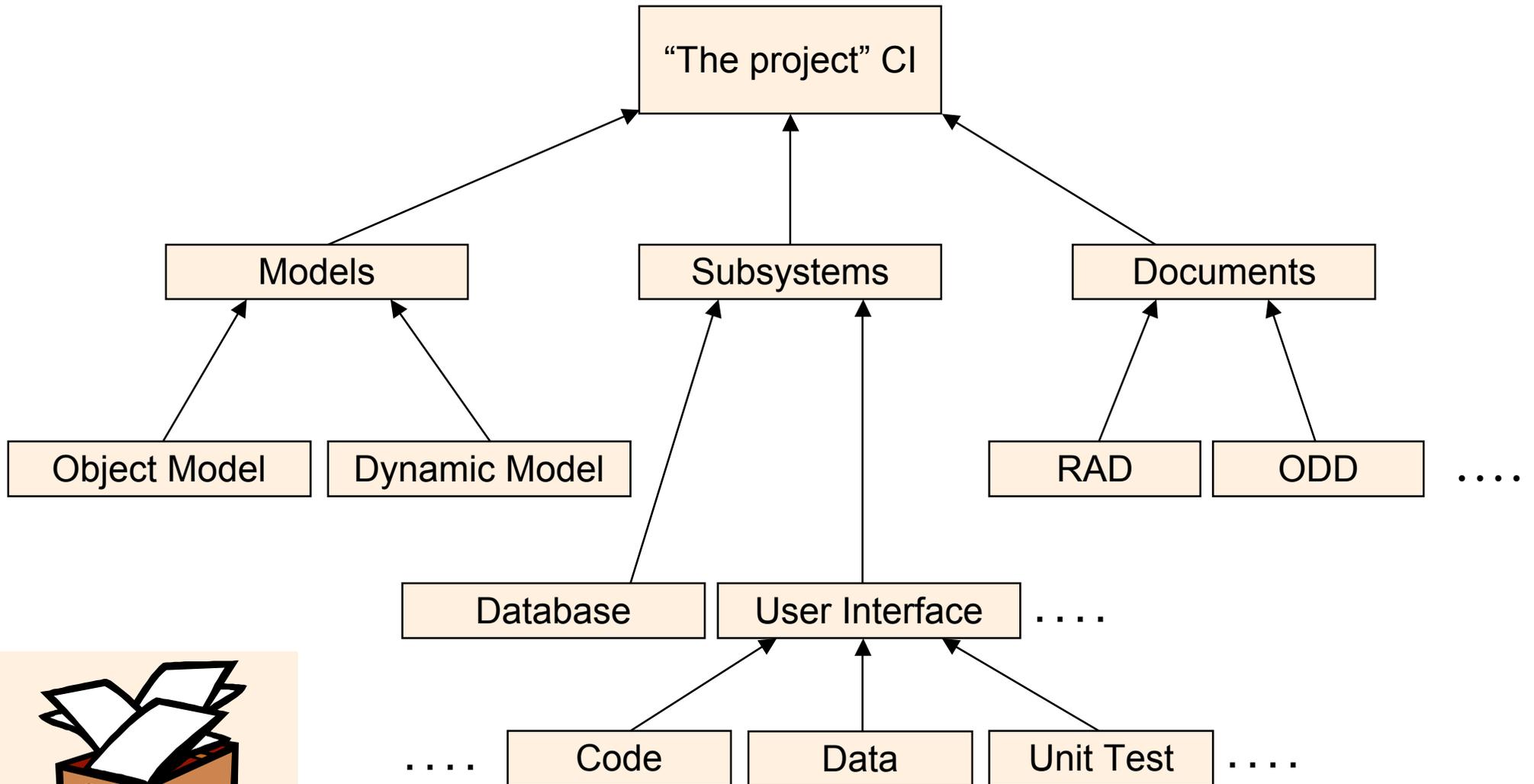
Terminology: Version

- The initial release or re-release of a configuration item associated with a complete compilation or recompilation of the item.

- Different versions have different functionality.

Terminology: Baseline

"A specification or product that has been formally reviewed and agreed to by responsible management, that thereafter serves as the basis for further development, and can be changed only through formal change control procedures."
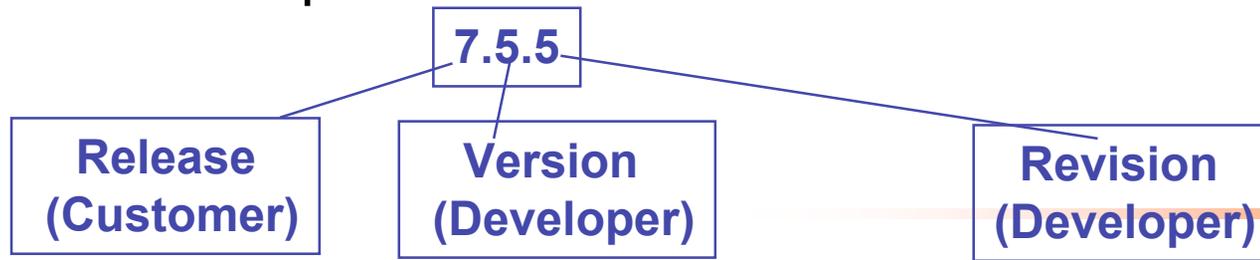
Examples:

Baseline A: All t**he API have completely been defined; the bodies of the methods are empty**.

Baseline B: **All data access methods are implemented and tested**.
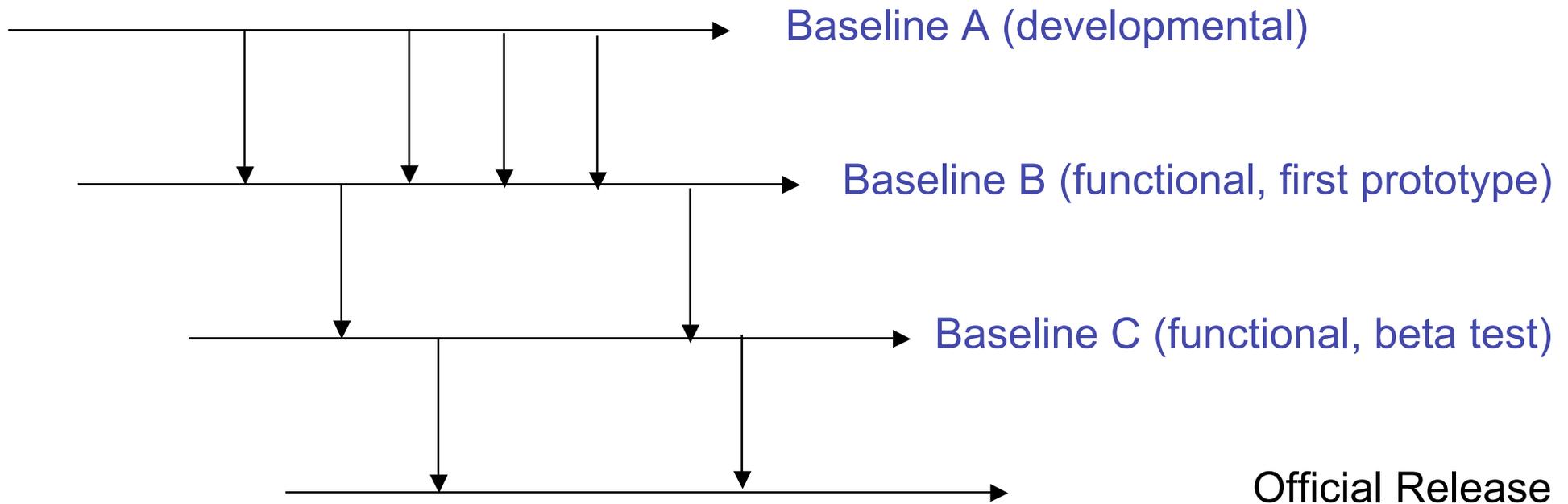
Baseline C: **The GUI is implemented.**

More on Baselines

- As systems are developed, a series of baselines is developed, usually after a review (analysis review, design review, code review, system testing, client acceptance, ...)
  - *Developmental baseline* (RAD, SDD, Integration Test, ...)
    - Goal: Coordinate engineering activities.
  - *Functional baseline* (first prototype, alpha release, beta release)
    - Goal: Get first customer experiences with functional system.
  - *Product baseline* (product)
    - Goal: Coordinate sales and customer support.
- Many naming scheme for baselines exist (1.0, 6.01a, ...)
- A 3 digit scheme is quite common:

7.5.5

| Release (Customer) | Version (Developer) | Revision (Developer) |

Baselines in SCM

Baseline A (developmental)

Baseline B (functional, first prototype)

Baseline C (functional, beta test)

Official Release
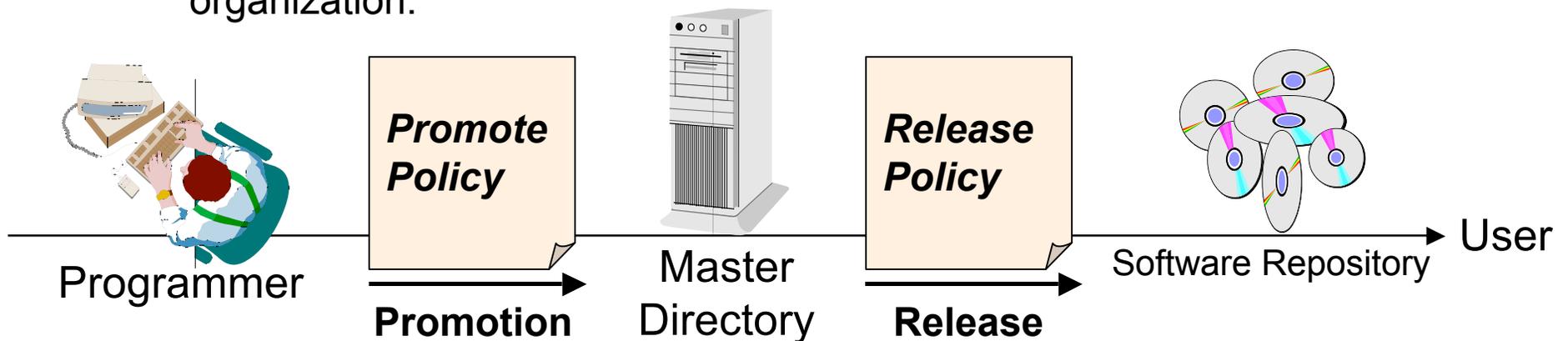
How do we manage changes in the baselines?

Time

Change management

- Change management is the handling of change requests
  - A change request leads to the creation of a new release
- General change process
  - The change is requested (this can be done by anyone including users and developers)
  - The change request is assessed against project goals
  - Following the assessment, the change is accepted or rejected
  - If it is accepted, the change is assigned to a developer and implemented
  - The implemented change is audited.
- The complexity of the change management process varies with the project. Small projects can perform change requests informally and fast while complex projects require detailed change request forms and the official approval by one more managers.

© 2004 B. Brügge, A. Dutoit, Pearson Education

Controlling Changes

- Two types of controlling change:
  - *Promotion:* The internal development state of a software is changed.
  - *Release:* A changed software system is made visible outside the development organization.

Programmer → **Promotion** → **Promote Policy** → **Master Directory** → **Release Policy** → **Release** → Software Repository → User

- Approaches for controlling change (Change Policy)
  - Informal (good for research type environments and promotions)
  - Formal approach (good for externally developed CIs and for releases)

© 2004 B. Brügge, A. Dutoit, Pearson Education
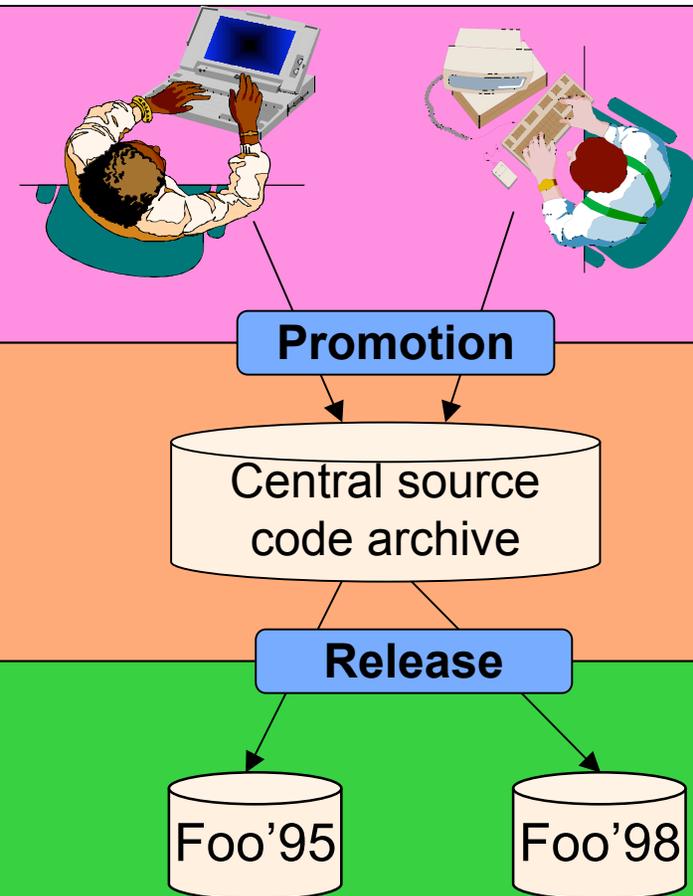
Terminology: SCM Directories

- **Programmer's Directory (IEEE: Dynamic Library)**
  - Library for holding newly created or modified software entities.
  - The programmer's workspace is controlled by the programmer only.

- **Master Directory (IEEE: Controlled Library)**
  - Manages the current baseline(s) and for controlling changes made to them.
  - Entry is controlled, usually after verification.
  - Changes must be authorized.

- **Software Repository (IEEE: Static Library)**
  - Archive for the various baselines released for general use.
  - Copies of these baselines may be made available to requesting organizations.
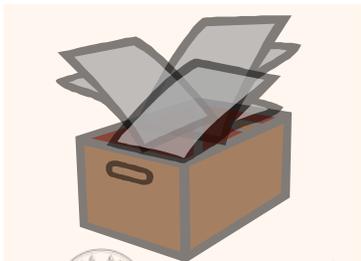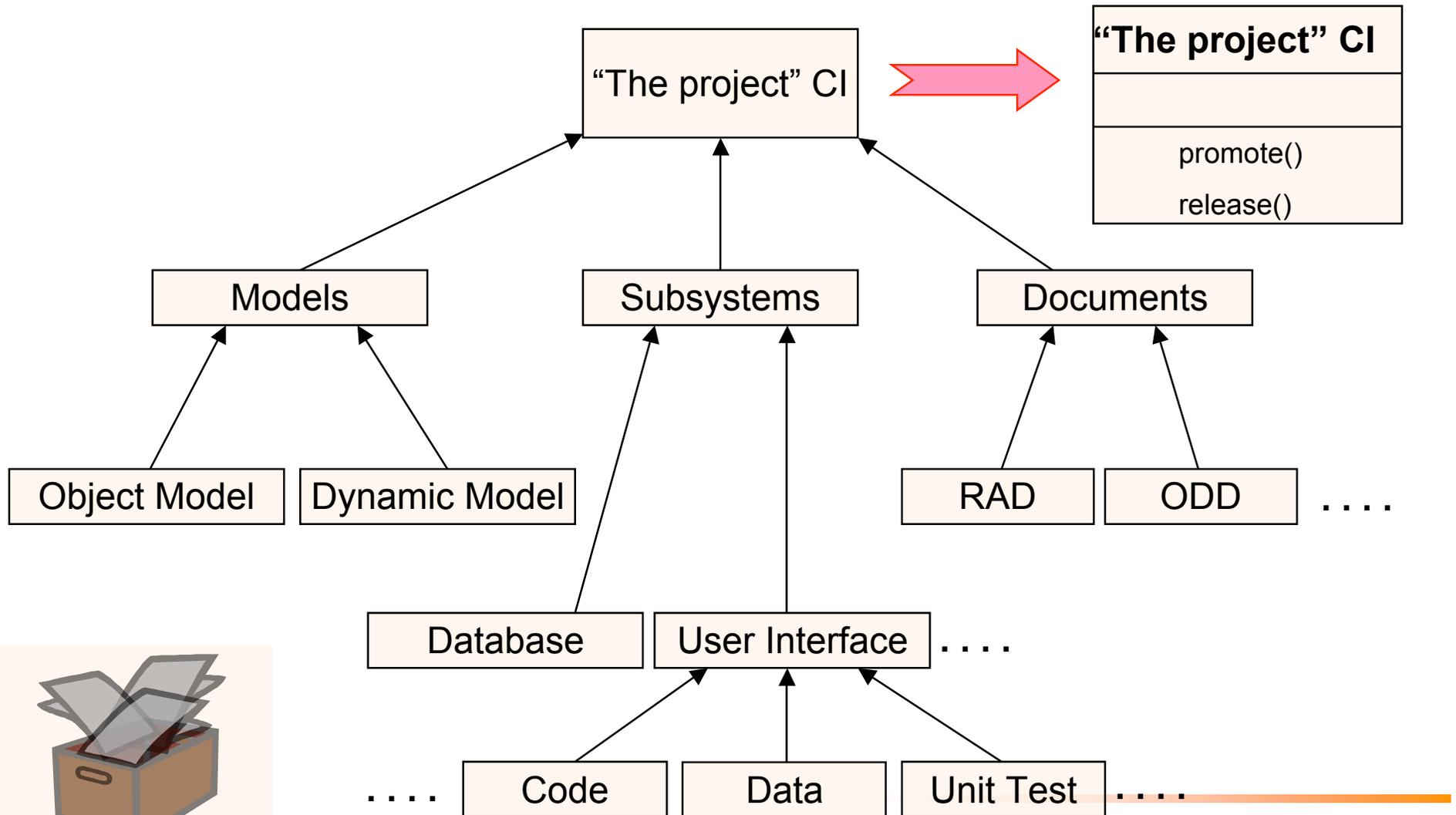
Standard SCM Directories

- **Programmer's Directory**
  - (IEEE Std: "Dynamic Library")
  - Completely under control of one programmer.

- **Master Directory**
  - (IEEE Std: "Controlled Library")
  - Central directory of all promotions.

- **Software Repository**
  - (IEEE Std: "Static Library")
  - Externally released baselines.

Promotion

Central source code archive

Release

Foo'95   Foo'98

© 2004 B. Brügge, A. Dutoit, Pearson Education

Promotion and Release are Operations on CIs

"The project" CI

"The project" CI
| |
| promote() |
| release() |

Models → Subsystems → Documents

Object Model   Dynamic Model

RAD   ODD   . . . .

Database   User Interface   . . . .

Code   Data   Unit Test   . . . .

. . . .

University of Zurich

"The project"

Let's Create a Model for Configuration Management

- We just learned that promotions are stored in the master directory and releases are stored in the repository
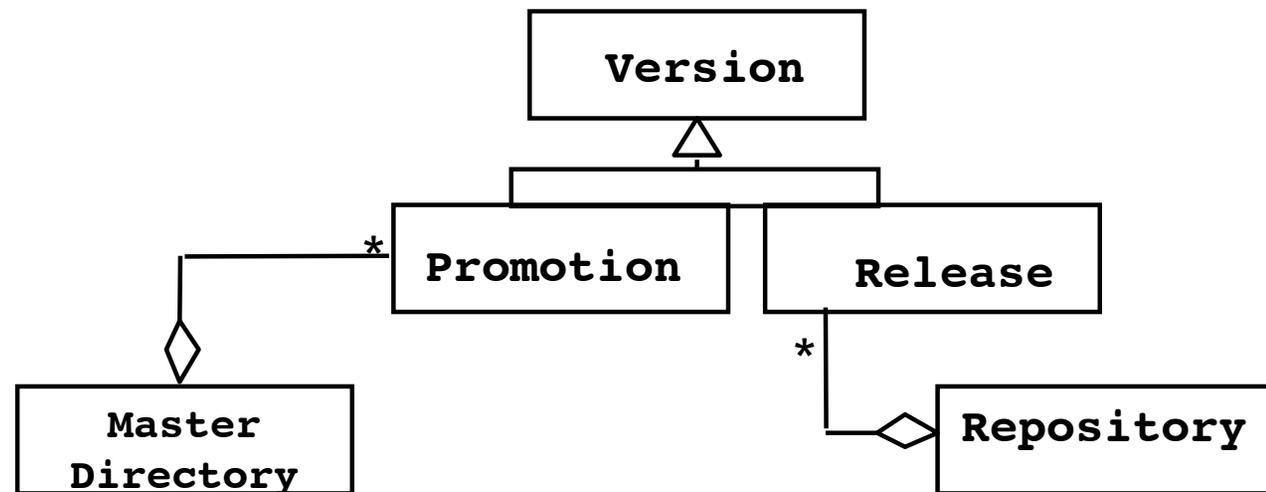
Problem: There can be many promotions and many releases
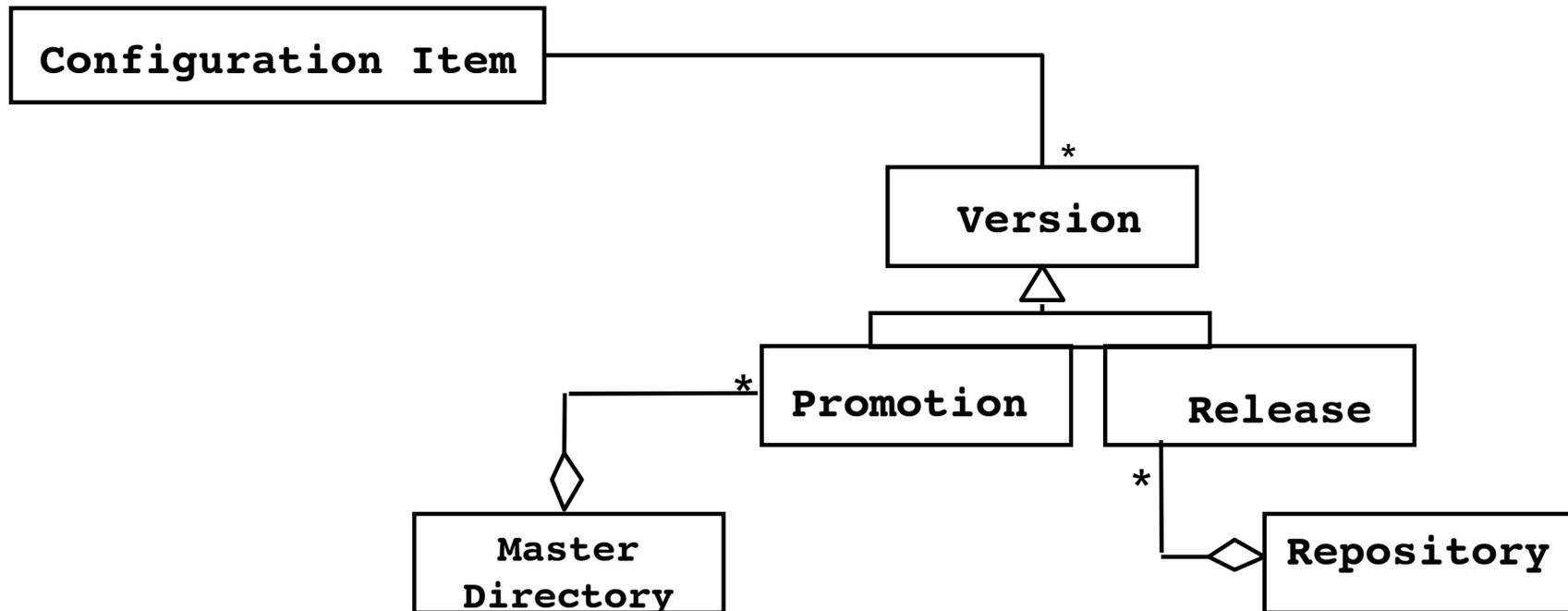Solution: Use Multiplicity

University of Zurich
Department of Informatics

Let's Create a Model for Configuration Management

■ Insight: Promotions and Releases are both versions

Solution: Use Inheritance

University of Zurich
Department of Informatics

Let's Create a Model for Configuration Management

- Problem: A configuration item has many versions

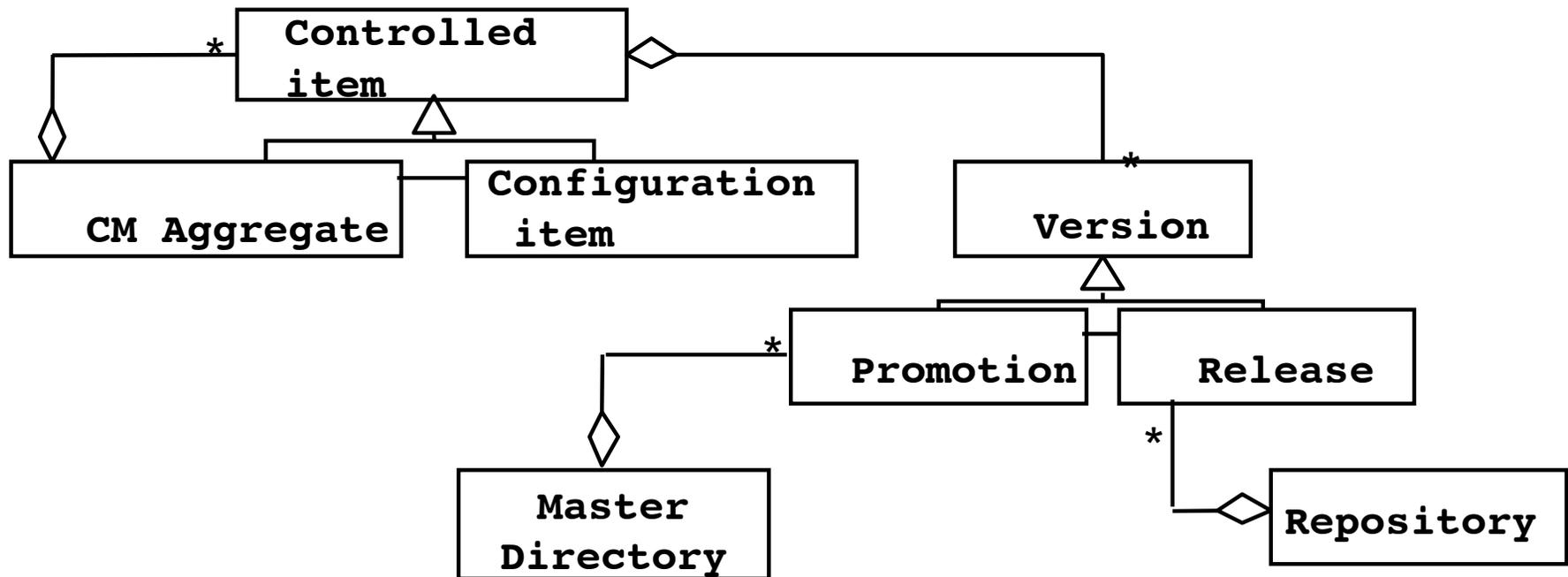Solution: Create a 1-many association between Configuration Item and Version

```
┌──────────────────────┐
│  Configuration Item  │────────────────┐
└──────────────────────┘                │
                                         │ *
                              ┌────────────────────┐
                              │      Version        │
                              └────────────────────┘
                                       △
                          ┌────────────┴────────────┐
                ┌──────────────┐          ┌──────────────┐
          * ────│  Promotion   │          │   Release    │
          │     └──────────────┘          └──────────────┘
          ◇                                       │ *
┌──────────────┐                          ◇──┌──────────────┐
│    Master    │                             │  Repository  │
│   Directory  │                             └──────────────┘
└──────────────┘
```

Let's Create a Model for Configuration Management

■ Problem: Configuration items can themselves be grouped
Solution: Use the composite design pattern
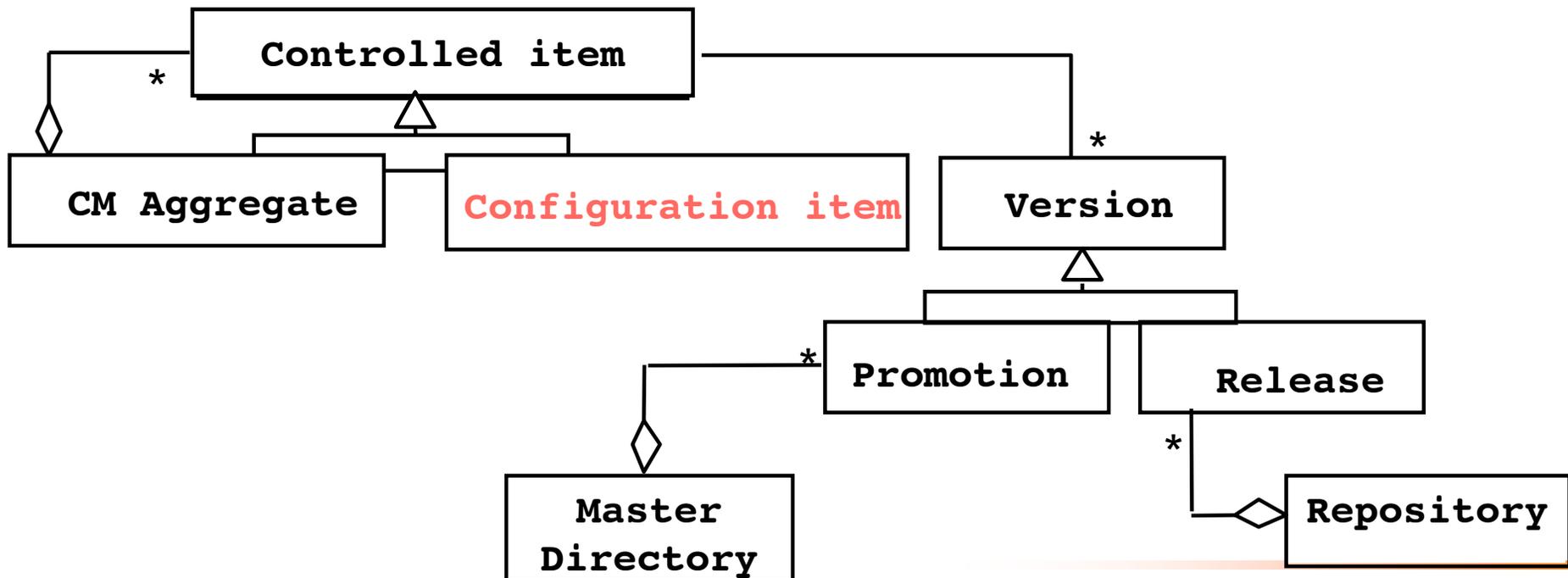
Configuration Item Model (UML Class Diagram)

Change Policies

- Whenever a promotion or a release is performed, one or more policies apply. The purpose of change policies is to guarantee that each version, revision or release (see next slide) conforms to commonly accepted criteria.

- Examples for change policies:
    - "No developer is allowed to promote source code which cannot be compiled without errors and warnings."

    - "No baseline can be released without having been beta-tested by at least 500 external persons."

Terminology: Version vs. Revision vs. Release

- ## Version
  - An initial release or re-release of a configuration item associated with a complete compilation or recompilation of the item. Different versions have different functionality.

> Question: Is Windows98 a new version or a new revision compared to Windows95 ?

- ## Revision
  - Change to a version that corrects only errors in the design/code, but does not affect the documented functionality.

- ## Release
  - The formal distribution of an approved version.

University of Zurich
Department of Informatics

Tasks for the Configuration Managers

Define configuration items

Define promote /release policies

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.

© 2004 B. Brügge, A. Dutoit, Pearson Education

The Software Configuration Management Plan

- Defines the *types of documents* to be managed and a document naming scheme.

- Defines *who takes responsibility* for the CM procedures and creation of baselines.

- Defines *policies for change* control and version management.

- Describes the *tools* which should be used to assist the CM 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.

University of Zurich
Department of Informatics

SCMP Section 1: Introduction

- 1.1 Simplified overview of the configuration management activities.
- 1.2 Scope:
  - Overview description of the project
  - Identification of the CI(s) to which software configuration management will be applied.
- 1.3 Identification of other software to be included as part of the SCMP (support software and test software)
- 1.4 Relationship of SCM to hardware of system configuration management activities
- 1.5 Degree of formality and depth of control for applying SCM to project.
- 1.6 Limitations and time constraints for applying SCM to this project
- 1.7 Assumptions that might have an impact on the cost, schedule and ability to perform defined SCM activities.

© 2004 B. Brügge, A. Dutoit, Pearson Education

SCMP Section 2: Management

- ## 2.1 Organization
  - Organizational context (technical and managerial) within which the SCM activities are implemented. Identifies
    - All organizational units (client, developers, managers) that participate in an SCM activity
    - Functional roles of these people within the project
    - Relationship between organizational units

- ## 2.2. Responsibilities
  - For each SCM activity list the name or job title to perform this activity
  - For each board performing SCM activities, list
    - purpose and objectives
    - membership and affiliations
    - period of effectivity, scope of authority
    - operational procedures

- ## 3. Applicable Policies
  - External constraints placed on the SCMP

SCMP Section 3: Activities

- 3.1 Configuration Identification
- 3.2 Configuration Control
- 3.3 Configuration Status Accounting
- 3.4 Configuration Audits and Reviews
- 3.5 Interface Control

© 2004 B. Brügge, A. Dutoit, Pearson Education

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

© 2004 B. Brügge, A. Dutoit, Pearson Education

3.2.1 Change Request

- Specifies the procedures for requesting a change to a baselined CI and the information to be documented:
    - Name(s) and version(s) of the CI(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

© 2004 B. Brügge, A. Dutoit, Pearson Education

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

- This section of the SCMP describes the organiztion 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.

- This section of the SCMP also indicates the level of authority of the CCB and its responsibility.
  - In particular, the SCMP must specify when the CCB is invoked.

© 2004 B. Brügge, A. Dutoit, Pearson Education

3.2.4 Implementing Change

- This section of the SCMP specifies the activities for verifying and implementing an approved change.
- A completed change request must contain the following 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 CIs to the configuration
    - How to deliver a new baseline

University of Zurich
Department of Informatics

3.3 Configuration Status Accounting

- This section of the SCMP must contain the following sections

  - 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

- This section of the SCMP 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 the Configuration Managers (Summary)

Define configuration items

Define promote /release policies

Define activities and responsibilities

Form of an SCMP

- ## Form:
  - The SCMP can be a separate document or a section embedded in another document, for example in the SPMP, titled "Software Configuration Management Plan".

- ## Minimum information
  - 6 Sections: Introduction, Management, Activities, Schedules, Resources and Plan Maintenance

- ## Consistency Criteria (to be used at a SCMP review meeting):
  - All activities defined in the SCMP (Section 3.1 to 3.6) are assigned to an organizational unit or person.
  - All identified Configuration items (Section 2.1) have defined processes for baseline establishment and change control (Section 3.2)
  - All activities are associated with resources (section 5) to accomplish the activities.

- ## Such a SCMP can include the following sentence:
  - "This SCM Plan conforms with the requirements of IEEE Std 828-1990."

University of Zurich
Department of Informatics

Tailoring the SCMP

- The IEEE standard allows quite a bit flexibility for preparing an SCMP.

- To conform to the rest of the project, 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.
    - Instead of omitting the associated section, mention its applicability.
    - Information that has not been decided on at the time the SCMP is approved should be marked as "to be determined".

Conformance to the IEEE Standard 828-1990

- Presentation format & Minimum information
  - A separate document or a section embedded in another document titled "Software Configuration Management Plan".
  - 6 Sections: Introduction, Management, Activities, Schedules, Resources and Plan Maintenance
- Consistency Criteria:
  - All activities defined in the SCMP (Section 3.1 to 3.6) are assigned to an organizational unit or person and they are associated with resources to accomplish the activities.
  - All Configuration items identified in Section 2.1 have defined processes for baseline establishment and change control (Section 3.2) .
- If the above criteria are met, the SCMP can include the following sentence:
  - "This SCMP conforms with the requirements of IEEE Std 828-1990."
- Note: The consistency criteria can also be used at a SCMP review meeting

Example SCM Plans (from the Guide IEEE 1042.1990)

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 softwae 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 applicatåions

Tools for Software Configuration Management

- Software configuration management is normally supported by tools with different functionality.
- Examples:
  - RCS
    - very old but still in use; only version control system
  - CVS (Concurrent Version Control)
    - based on RCS, allows concurrent working without locking
    - http://www.cvshome.org/
    - CVSWeb: Web Frontend to CVS
  - 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/

University of Zurich
Department of Informatics

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)

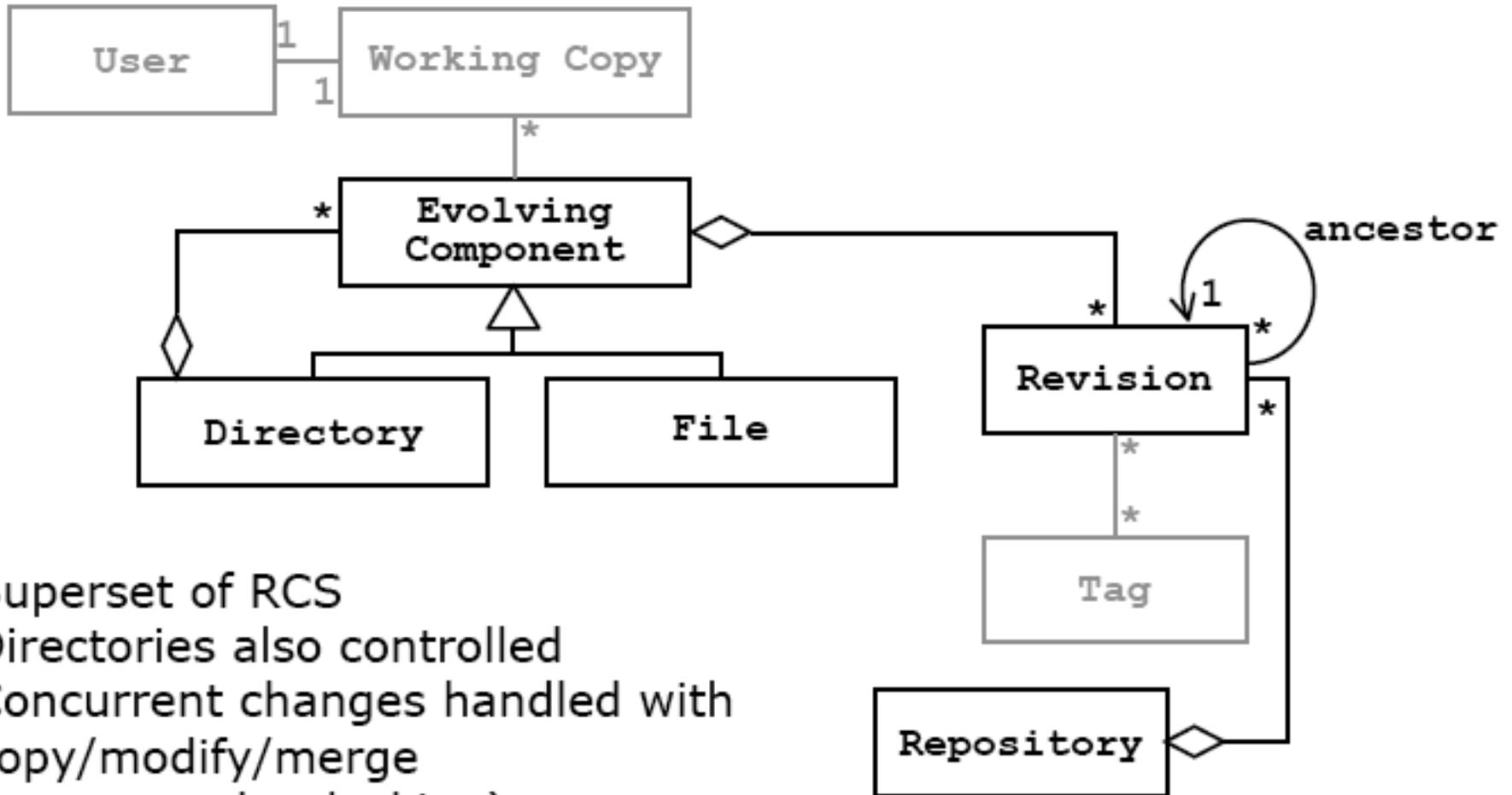# Full Model of SCM
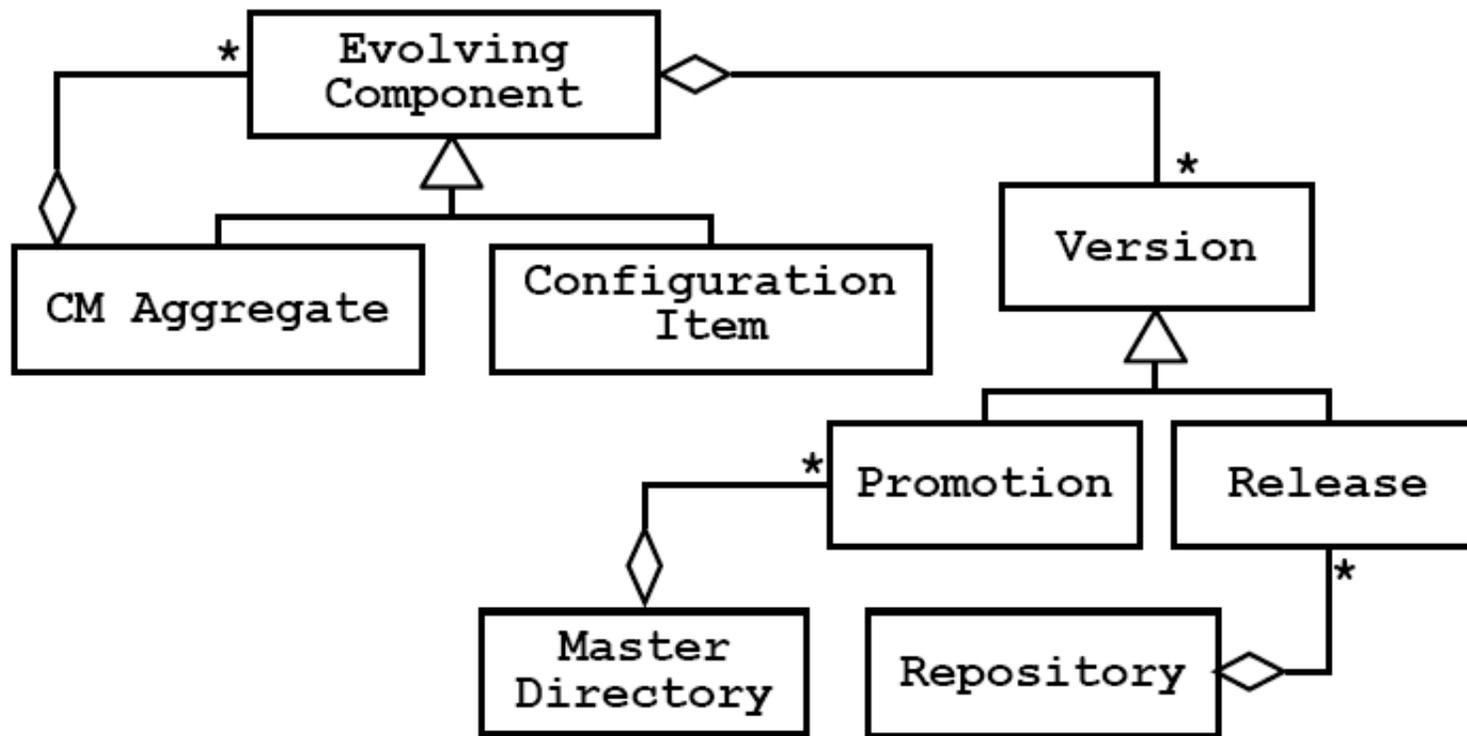
RCS concepts
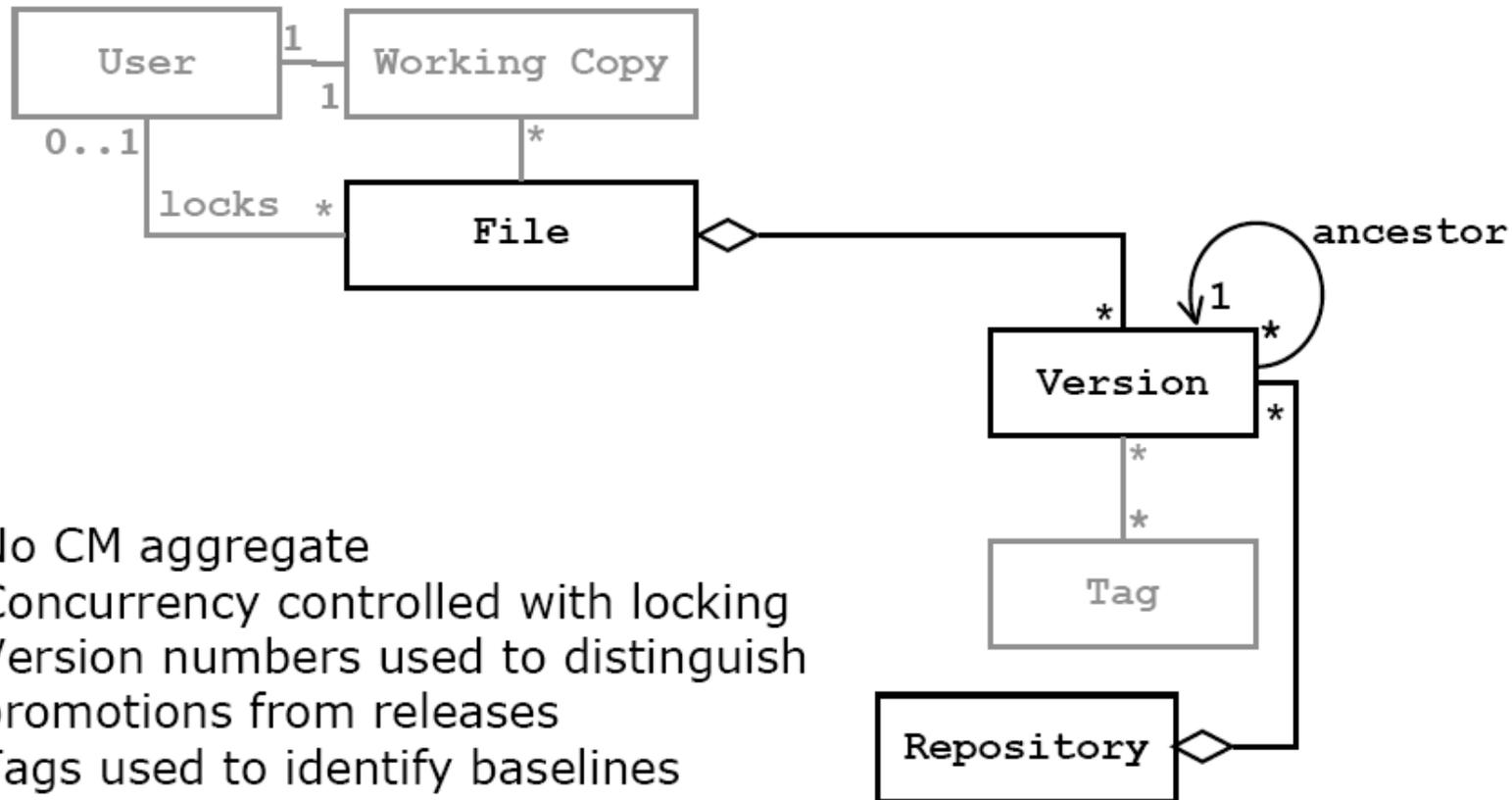
No CM aggregate
Concurrency controlled with locking
Version numbers used to distinguish
promotions from releases
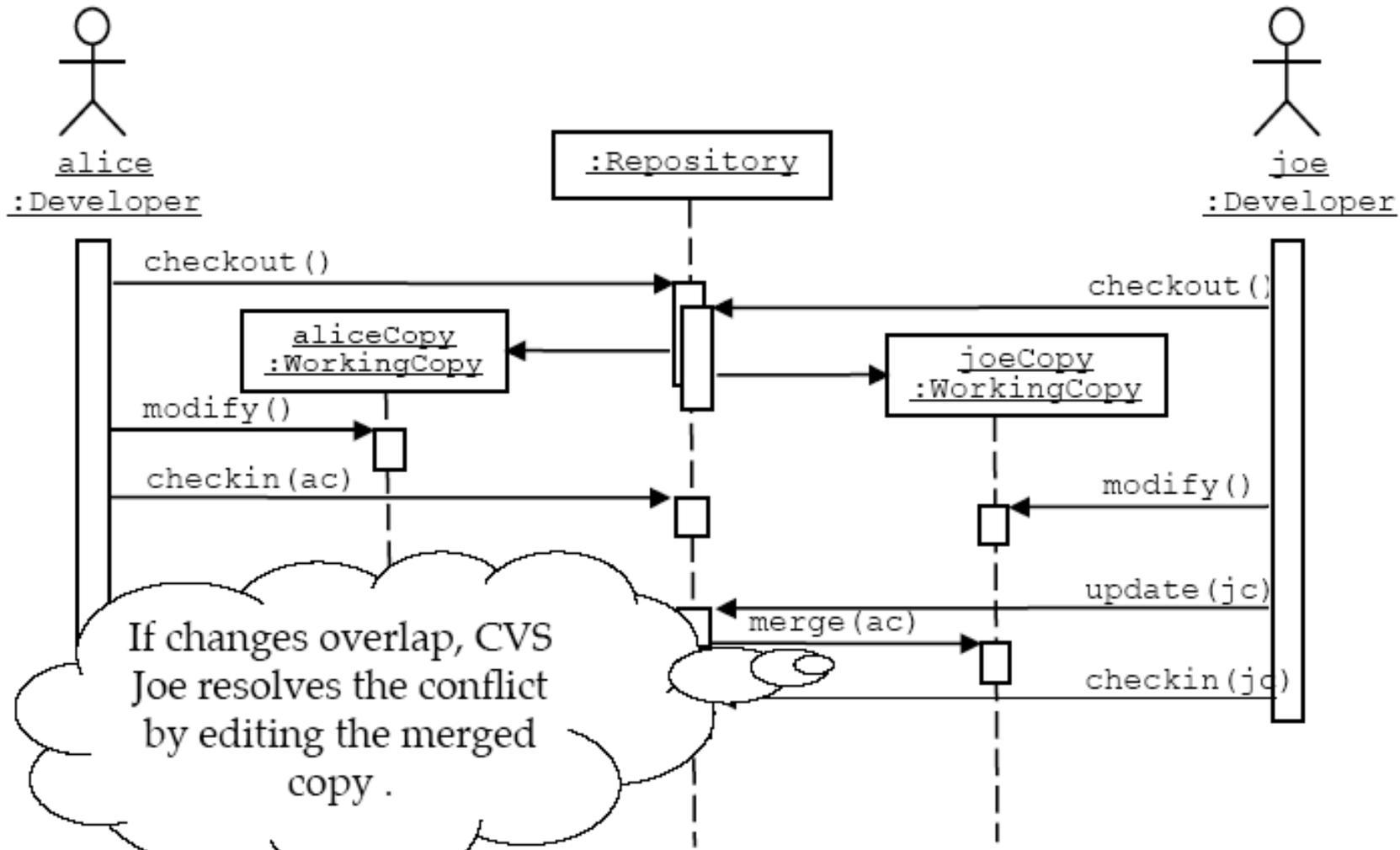Tags used to identify baselines

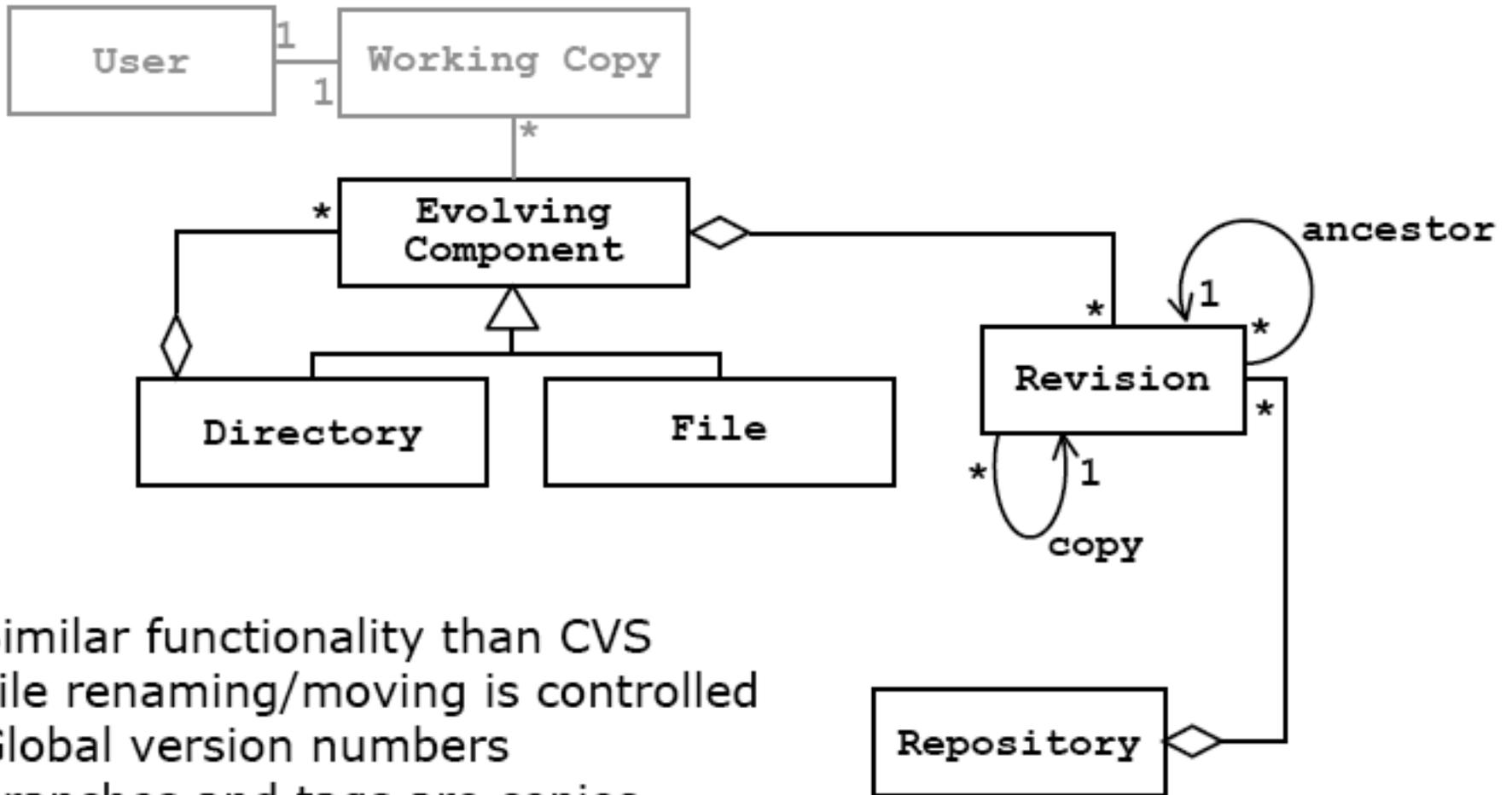# RCS numbering
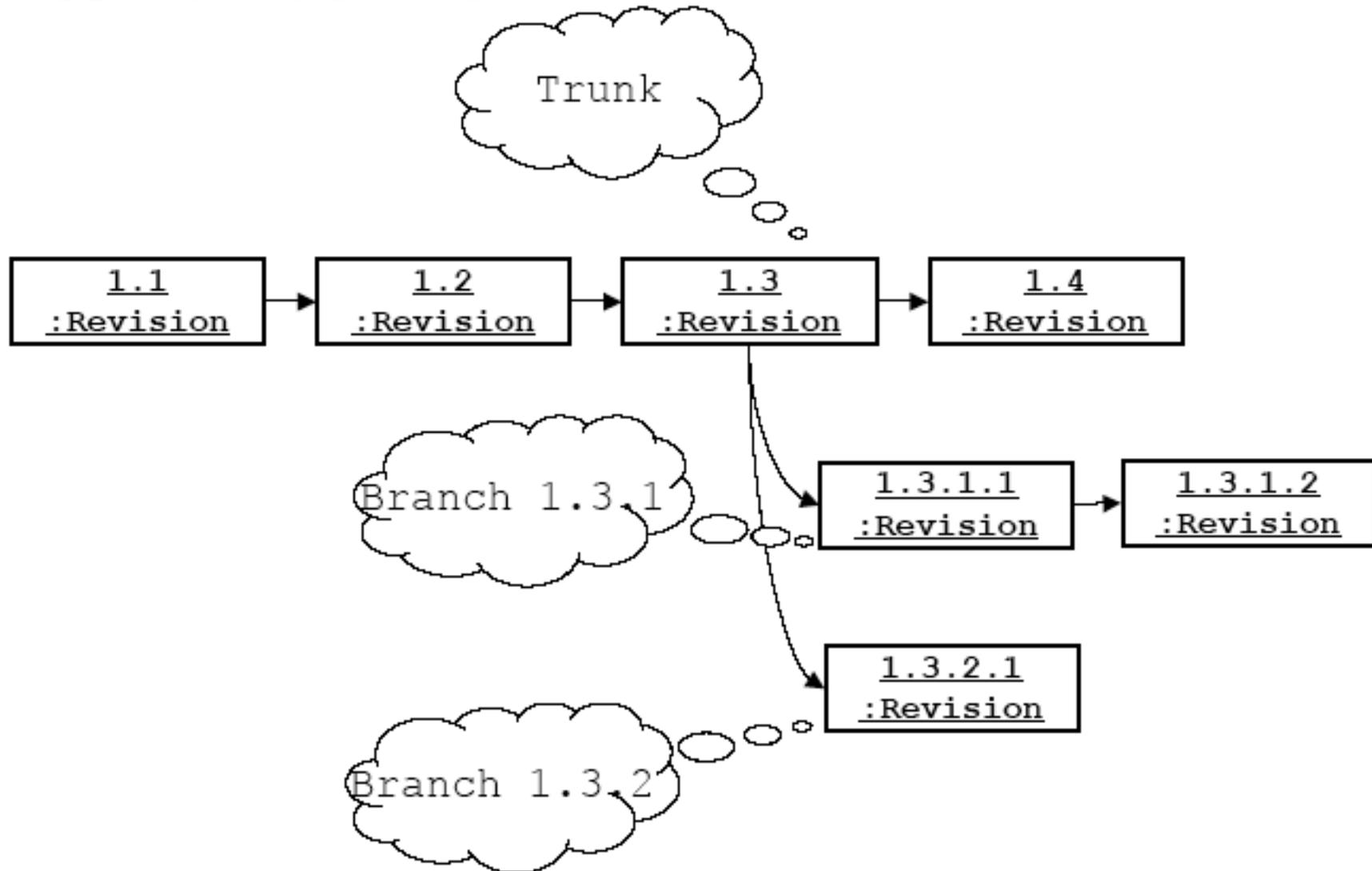
CVS concepts

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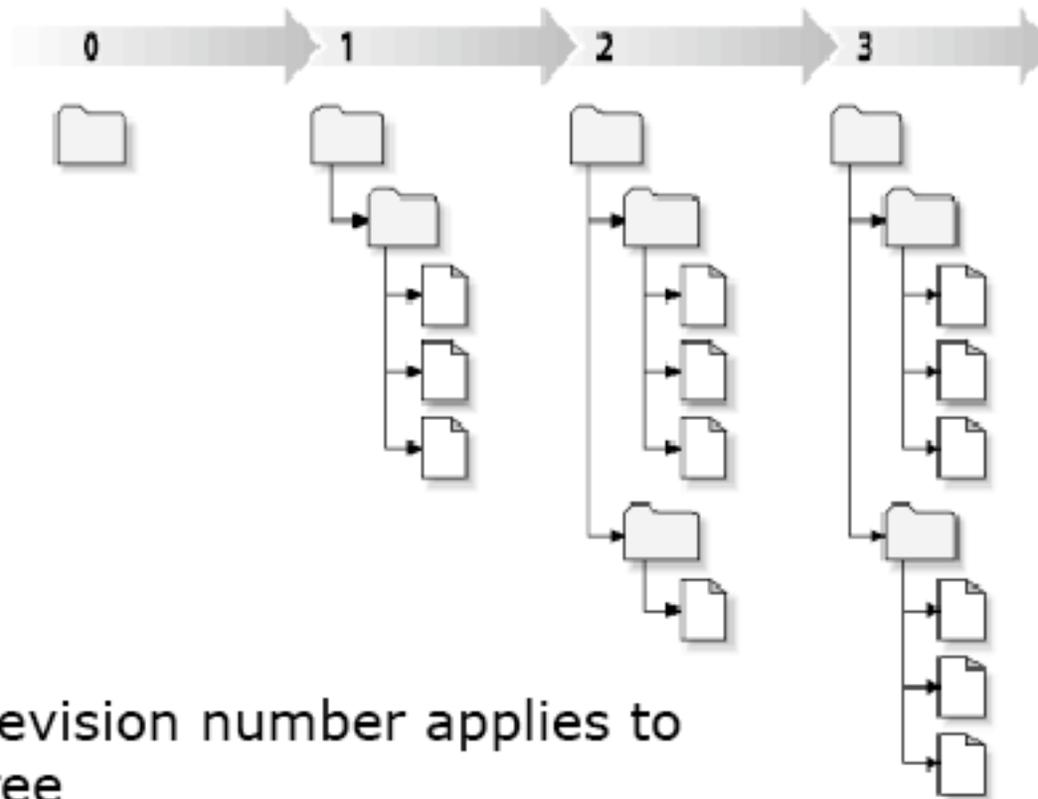(as opposed to locking)

CVS copy/modify/merge

SVN concepts

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

University of Zurich
Department of Informatics

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.

Tasks for the Configuration Managers

SCMP following the IEEE 828-1990 standard

Define configuration items

Define promote /release policies

Define activities and responsibilities

Set up configuration management system

© 2004 B. Brügge, A. Dutoit, Pearson Education

References

- Readings used for this lecture
  - [Bruegge-Dutoit] Chapter 13 Configuration Management
  - [IEEE Std 828] Software Configuration Management
  - [IEEE Std 1042] Guide to Configuration Management Plan (SCMP)
- Additional References
  - CVS
    - Homepage: http://www.cvshome.org/
    - Online Documentation: http://www.cvshome.org/docs/manual/cvs.html
  - Jikes: Open Source Java Compiler maintained with CVS
    - Source tree (read only): http://sourcery.org/jikes/anoncvs.html
    - Jikes project portal http://sourcery.org/jikes
  - CVSWEB example
    - http://stud.fh-heilbronn.de/~zeller/cgi/cvsweb.cgi/

University of Zurich
Department of Informatics

Summary

- SCM: Important part of project management to manage evolving software systems and coordinate changes to them.
- SCM consists of several activities:
  - Promotion and Release management
  - Branch, Variant and Change Management
- Public standard for SCM plans: IEEE 828.
- The standard can be tailored to a particular project:
  - Large projects need detailed plans to be successful
  - Small projects should not be burdened with the bureaucracy of detailed SCM plans
- SCM should be supported by tools. These range from
  - Simple version storage tools
  - Sophisticated systems with automated procedures for policy checks and support for the creation of SCM documents.

University of Zurich
Department of Informatics