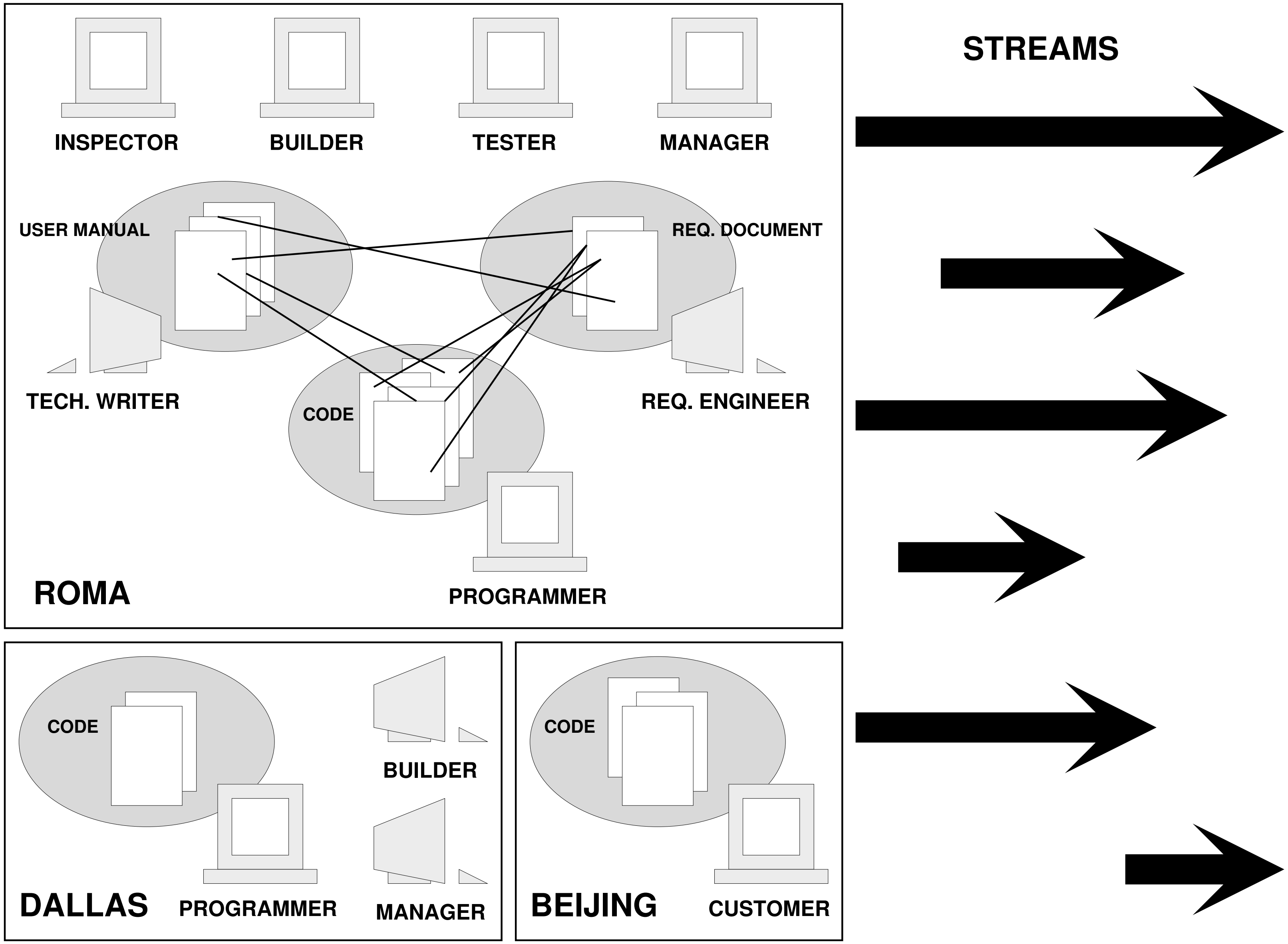
SOFTWARE CONFIGURATION MANAGEMENT

Alexander L. Wolf

Department of Computer Science
University of Colorado at Boulder
Boulder, Colorado 80309 USA

WHAT IS THE STATE OF THE ART?

- *THE PROBLEM*
- *ISSUES*
- *APPROACHES*
- *RESEARCH CHALLENGES*

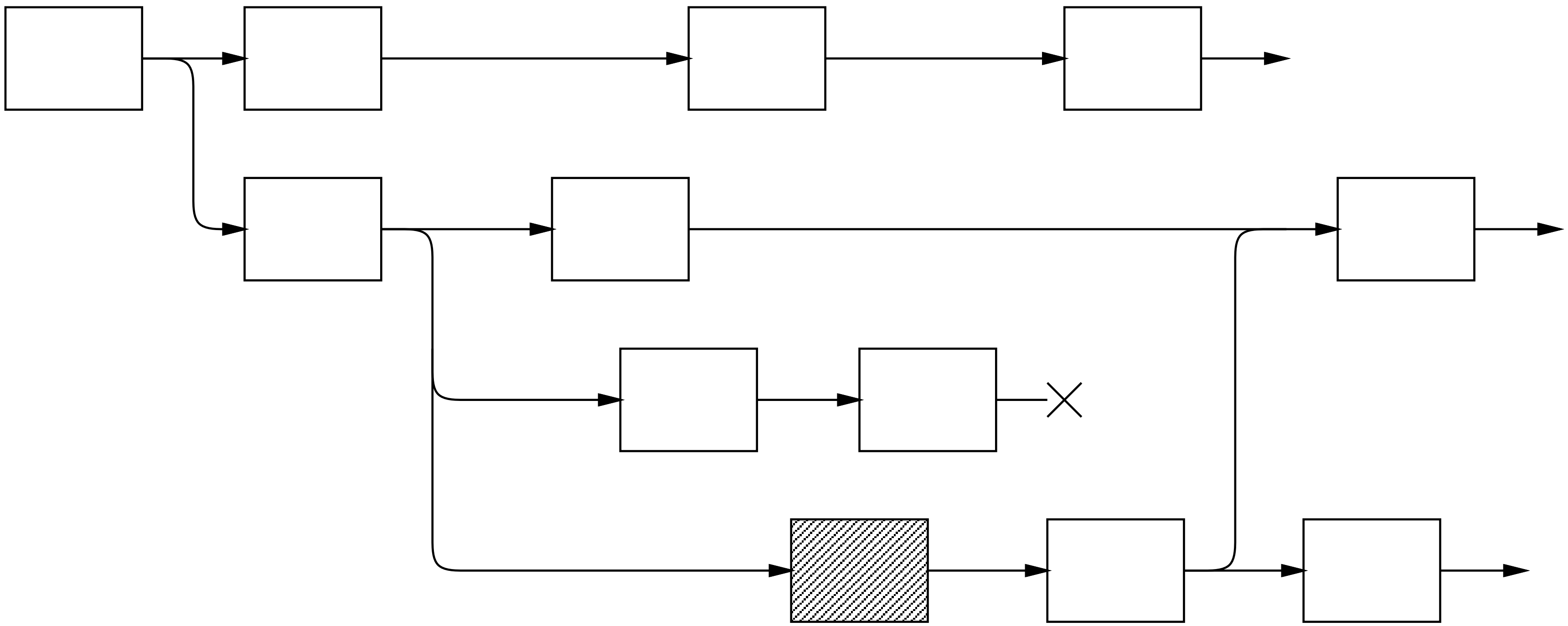


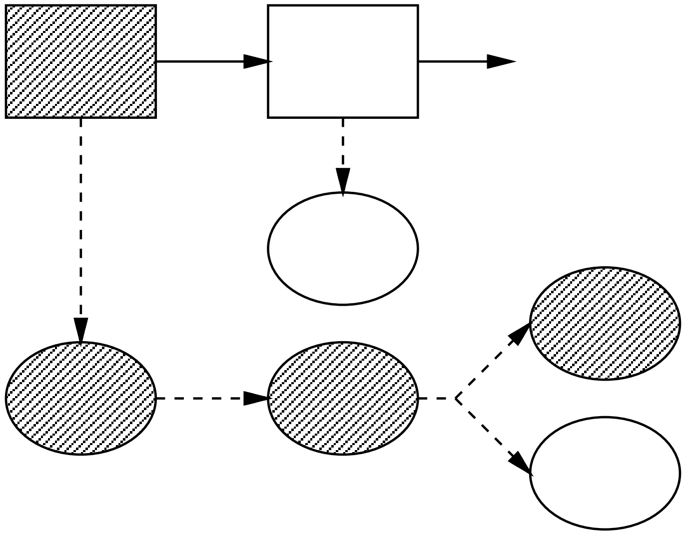
INSPECTOR
BUILDER
TESTER
MANAGER
USER MANUAL
REQ. DOCUMENT
TECH. WRITER
CODE
REQ. ENGINEER
ROMA
PROGRAMMER
CODE
BUILDER
DALLAS
PROGRAMMER
MANAGER
CODE
BEIJING
CUSTOMER
STREAMS

A DEFINITION

Software Configuration Management (SCM) is the disciplined archiving, composition, and evolution of the artifacts associated with a software system.

- *IDENTIFICATION*
- *VERSION SELECTION*
- *CHANGE CONTROL*
- *MANUFACTURE/REMANUFACTURE*
- *STATUS ACCOUNTING*
- *AUDIT/REVIEW/ANALYSIS*





Configuration

Derived and Source Items

Version Graph

Merge

Variant (Branch)

Revision (Fix)

Configuration Item

COMPONENTS OF A SOLUTION

- *REPOSITORY*
  $\Leftarrow$ identification, (re)manufacture, status accounting, audit/review/analysis
- *PROCESS MODEL*
  $\Leftarrow$ change control, (re)manufacture, audit/review/analysis
- *SYSTEM MODEL*
  $\Leftarrow$ version selection, change control, (re)manufacture

COMPLICATIONS

- *RELATIONSHIP TO COMPONENT STRUCTURE*

  Software architecture

- *RELATIONSHIP TO PROJECT STRUCTURE*

  Teams working concurrently

  Geographic distribution

  Virtual corporation

- *RELATIONSHIP TO OTHER PROCESSES*

  Bug tracking, field support, product planning

- *LIMITED RESOURCES*

  Automation

  Optimization (space and time)

SO, WHAT IS THE STATE OF THE ART?

- *PROCESS PERSPECTIVE: HOW?*
- *TECHNOLOGY PERSPECTIVE: WITH WHAT TOOLS?*

PROCESS PERSPECTIVE

- *MARKET DEPENDENT*

  (e.g., Microsoft vs. AT&T)

- *PRODUCT-MATURITY DEPENDENT*

  (e.g., pre-fielding vs. post-fielding)

- *EGO DEPENDENT*

  (e.g., customer vs. corporation vs. project vs. developer)

CLOSER LOOK: APPROACHES TO DEVELOPER SCM

[P. Feiler, Software Engineering Institute]

- *CHECK-OUT/CHECK-IN*

  Example: Unix RCS

- *COMPOSITION*

  Example: Apollo DSEE

- *NON-SERIALIZABLE TRANSACTIONS*

  Example: Sun NSE

- *CHANGE SET*

  Example: SMDS Aide-De-Camp

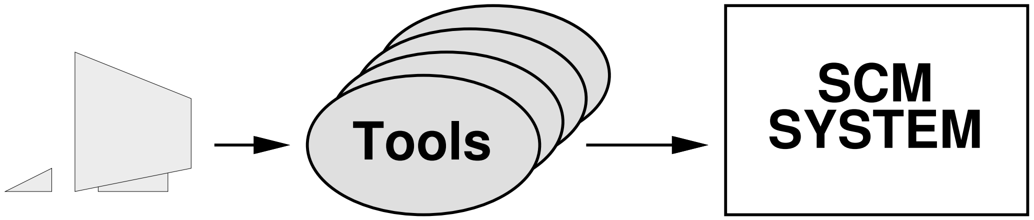
TECHNOLOGY PERSPECTIVE

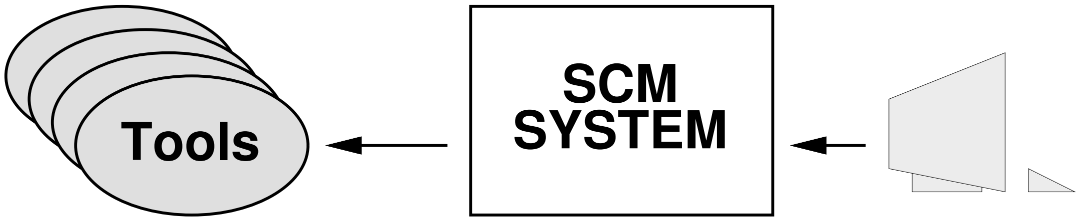
- *IN-HOUSE DEVELOPMENT*
  - \+ Tailored to local environment
  - – Tends to be inflexible
- *INDEPENDENT VENDORS*
  - \+ Strong conceptual model
  - ○ Management/control oriented
  - – Hard to integrate with local environment
- *ACADEMIC AND INDUSTRIAL RESEARCH*
  - \+ General-purpose, flexible solutions
  - ○ Developer oriented
  - – Often does not scale

STATE OF THE ART IN SOLUTION COMPONENTS

- *REPOSITORY*
  - Objects, relationships, and operations
  - Beyond serializability
  - Local-area distribution
  - Derived object (item) pools and delta storage
- *PROCESS MODEL*
  - Formalisms and environments
  - Workspaces and concurrency
- *SYSTEM MODEL*
  - Formalisms
  - Consistency constraints (truth maintenance)

CLOSER LOOK: SCM AND TOOLS





CLOSER LOOK: 3-D FILE SYSTEM

- *ADDS NEW TYPE OF FILE*

  Collects file versions under same name

- *AKIN TO DIRECTORIES*

  Contains entries for **version instance files**

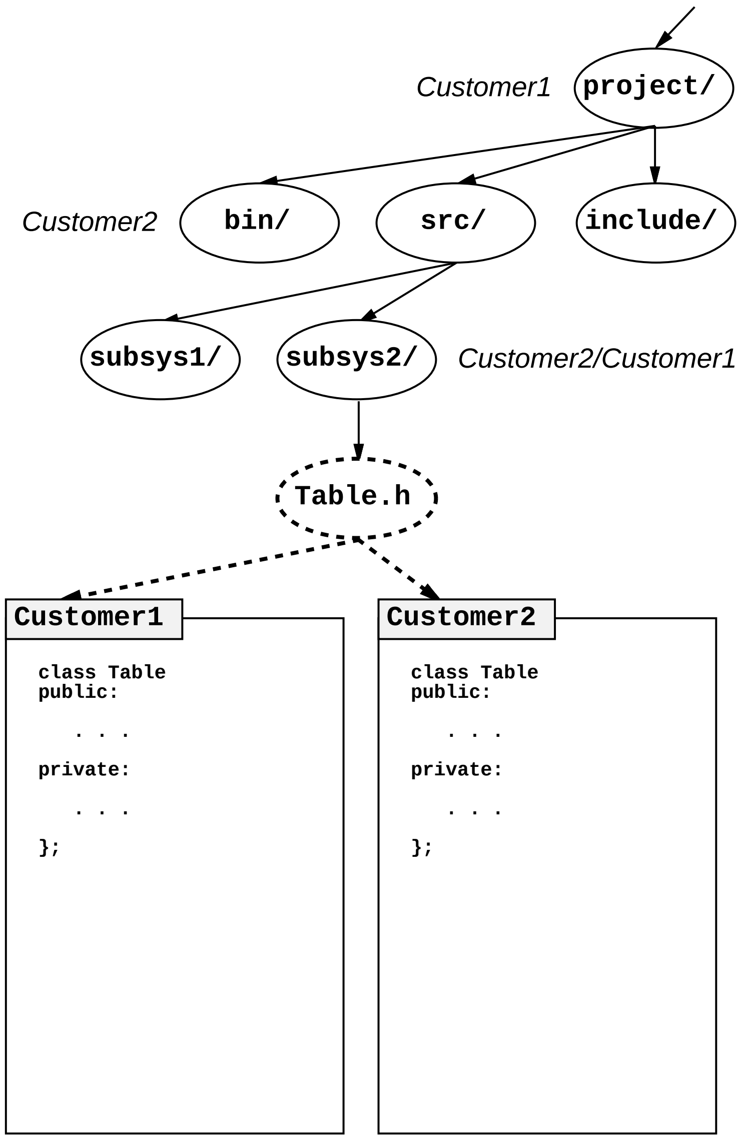
  But... reference to version file is reference to instance, not collection of instances

- *VERSION SELECTION*

  References filtered through per-process **mapping table**

- *MECHANISM, NOT POLICY*

  e.g., applications give names to instances



VERSION MAPPING TABLE

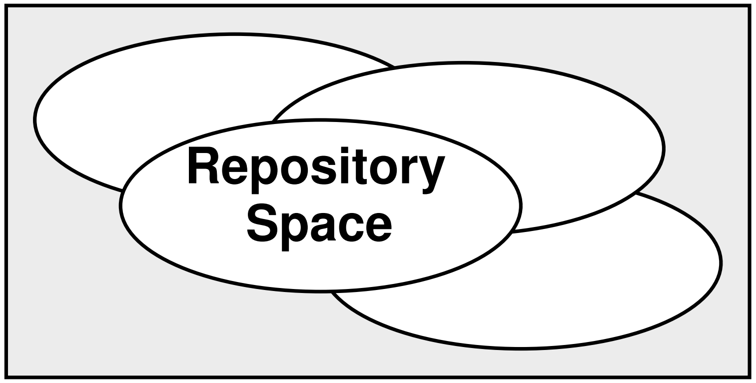
DIRECTORY	VERSION
. . ./project	*Customer1*
. . ./project/bin	*Customer2*
. . ./project/src/subsys2	*Customer2/Customer1*
. . .	. . .

CLOSER LOOK: SCM WITH ADELE

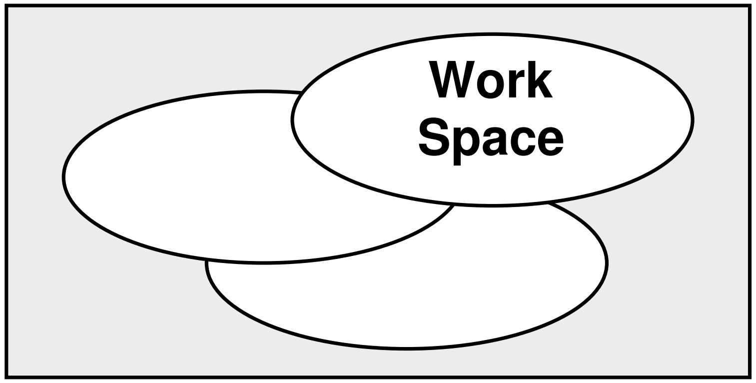
- *SOFTWARE ENGINEERING ENVIRONMENT*

  Data management, tool integration, and process support

- *OBJECT AND PROCESS MANAGEMENT*



Object Manager



Process Manager

ADELE REPOSITORY SPACE

- *E-R-A DATA MODEL*
- *BRANCHING VERSION MODEL*
- *CONFIGURATION MODEL VS. INSTANCE*
- *FIRST-ORDER LOGIC CONSTRAINTS ON ATTRIBUTES*

ADELE WORK SPACE

- *WORK SPACE AS SUB-DATABASE*
- *ALLOWS STANDARD TOOLS (FILE MODEL)*
- *EVENTS/CONDITIONS/ACTIONS ⇒ TRIGGERS*

RESEARCH CHALLENGES

- *MERGING AT DEEP SEMANTIC LEVEL*
  - Feed field changes into development streams
  - Propagate bug fixes
- *EVOLVING COMPLEX OBJECTS*
  - Relationships
- *SCALE*
  - Size of projects
  - Number of streams
  - Number of components
  - Geographic dispersion

RESEARCH CHALLENGES

- *VIRTUAL CORPORATIONS*

  Process integration

  Selective information sharing

- *SUPPORT FOR NEW MODELS OF SOFTWARE DEVELOPMENT*

  Software composition

FURTHER INFORMATION AND DETAILS

- *SCM INTERNATIONAL WORKSHOP SERIES*

  Grassau, Germany (1988); Princeton, New Jersey (1989); Trondheim, Norway (1991); Baltimore, Maryland (1993); Seattle, Washington (1995)

- *SOFTWARE ENGINEERING INSTITUTE*

  Dart; Feiler; . . .

- *NUMEROUS BOOKS*

  Bersoff et al. (1980); Babich (1986); Arthur (1988); Whitgift (1991); . . .

- *NUMEROUS STANDARDS*

  DOD; IEEE; NASA; . . .