



Software Configuration Management

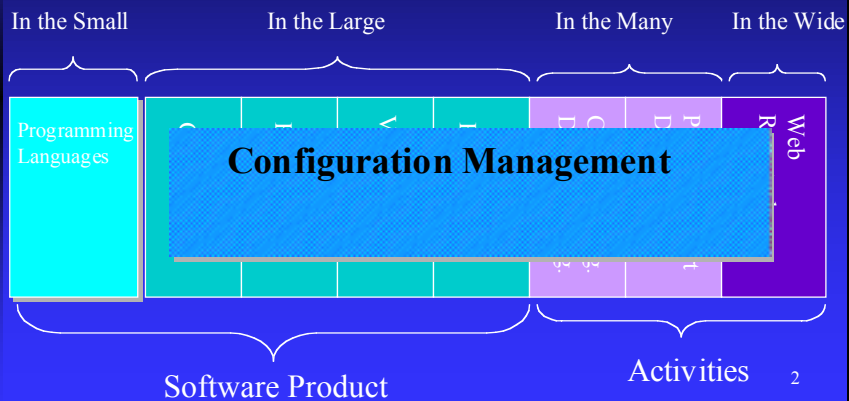
A Road Map

Jacky Estublier

Grenoble University - France



Software Engineering, Programming, and SCM





What is CM

CM is the **CONTROL** of the **EVOLUTION** of **COMPLEX** systems

Any complex system has :

- ◆ Large number of components and their versions
- ◆ Large number of people involved (up to 1000)
- ◆ Very hard time constraints (cycle down to 4 month)
- ◆ Long product life duration (10 to 20 years)

Goal: Keep evolving product under control and help satisfy delays and quality constraints

3



Configuration Management

Managing a repository of components

- Product and version Management

Help engineers in their usual activities

- Work Space and tool control

Process control and support

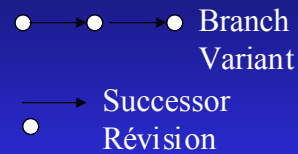
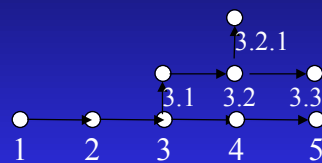
- Change control & Cooperative work

4



Component repository: Version control

early 70s



- History management
- Delta (space saving)
- Multi-user “control”
- Merging

Still the base of almost all SCM systems.

5



Versioning principles

Question: Why are 2 objects versions of each other

- ◆ They have something in common (but what?)
- ◆ They are different (in what?).

With current tools the answer is:

- ◆ They share a large number of lines,
- ◆ They have some lines which are different

6



Version Set

- A group of changes has a logical name (BugFix21)
- A specific release R2 can be built from a previous one R1
 - ◆ $R2 = R1 + \text{BugFix21} + \text{Extention2} - \text{Extention3}$
- We can build versions that never existed before !
- Higher conceptual level, easier to understand ...
- For long presented as the opposite to the version tree
- Is becoming available in new tools, with a version tree.

7



Component repository: Product models

Data Model : files and directories

We need a good DB one (incl. Relationships).

Product Model : System Model (80s) :

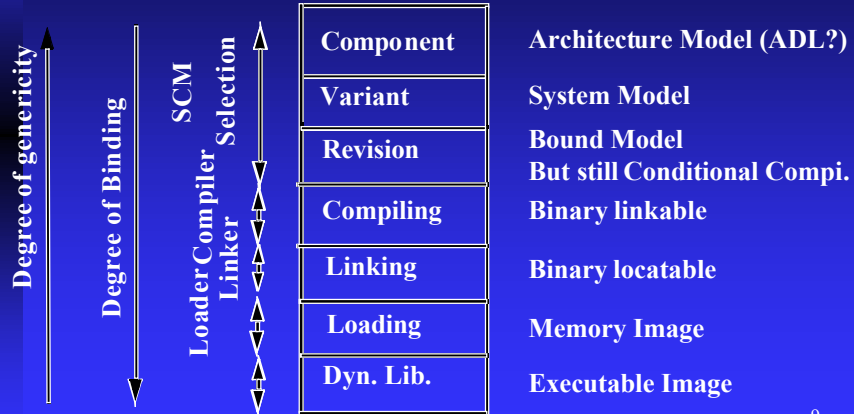
architecture / composition of a configuration :
Relationship between components, dependencies,
compatibility constraints, ... (MIL, ADL).

Most often only a Makefile contains that information!!.

8



Help engineers: Selection/Building



9



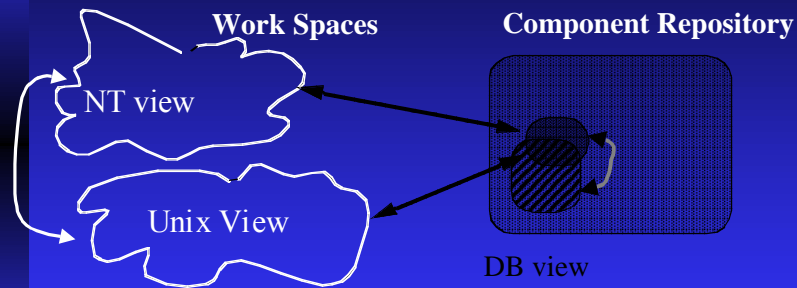
	ADL Software Architecture	SCM Conf. Management	CDS Conf. Distr. Syst.
Composition	ooo	o	oo
Behaviour	o		
Consistency	oo	o	o
Building	o	ooo	o
Versioning		ooo	o
Selection		oo	

Convergence, intégration, complement, interoperability ??.

10



Help engineers: Work Space Control



Problem: Define and control each arrow:
Concurrent Engineering control

11



Current Limitations

A **central** data base as repository.
=> Severe scalability and efficiency limitations.

Too low level product models
Single rigid projection.

Concurrent work definition and control
=> No explicit definition, No high level concepts
=> No way to enforce policies
=> Strong limitations in distributed work

12



Process Control and Support

Good knowledge and control of the software process is mandatory for quality, cost and delay predictability.

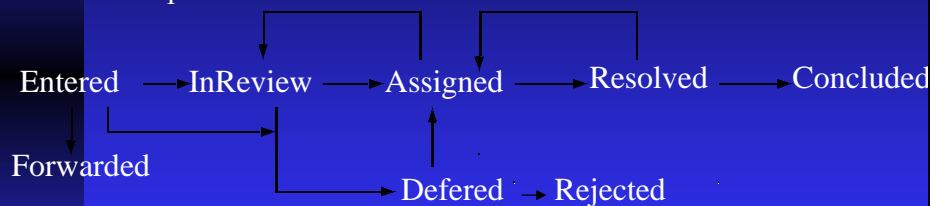
- ◆ Change Control.
Well identified and formalized,
=> Each SCM tool proposes a solution.
- ◆ Concurrent engineering
Team coordination and synchronization.
=> No satisfactory proposition.
- ◆ Others : Business, administration, organization
=> Weak and overlaps with other tools.

13



State of Practice in SCM

A process is described by a “State Transition Diagram” (STD) of the product.



It is a fine grained approach, but too low level.

14



Global assessment

- Strength : A successful technology because ..
 - ◆ It does not take into account the syntax and semantics of programming languages, applications, ...
 - ✓ => Simple, generic and efficient tools.
 - ✓ => Heuristics proved very effective.

SCM is a pragmatic approach to solve SE problems.

- Weakness
 - ◆ It does not take into account the syntax and semantics of programming languages, applications, ...
 - ✓ => No correctness criteria.
 - ✓ => Functionalities are limited.

15



Challenges 1 : Interoperability

A growing number of tools include :

- ◆ a data repository, versioning, concurrency control
Example : most 4GL tools, PDM tools, editing tools
- ◆ process support
Example : MS Project, Lotus, workflow, ...

And the increasing number of COTS tools, the increasing need to make interoperate tools /activities.

SCM tools must be designed as a component which interacts with many others, both for data AND process control reasons.

16



Challenge 2 : PDM v.s. SCM

- The increased part of software in any industrial product
- The need to develop consistently software and hardware
- Increasing overlap / conflict with PDM tools
 - ◆ Many similarities but
 - ◆ Many differences ...
- None can pretend to do both well.
- There is a need to reconcile these worlds. How ?
 - ◆ Integration
 - ◆ Cooperation
 - ◆ Extension ...

17



Challenges 3 : Remote development

The gap between local, distributed and remote development should (almost) disappear.

The virtual enterprise (distributed and heterogeneous) rises now problems.

Issues :

Efficiency, scalability, remote concurrent engineering control, heterogeneity, ...

18



Challenges 4 : New domains

New applications : ex Managing web pages

Potentially a larger market than Soft development

New / different issues

more pages (x10, x100)

faster changes (daily)

more (undisciplined) contributors (x100)

contributors are not soft. Professionals

Who will take that market : Document management tools,
SCM tools, others?.

19



Challenges 5 : Complexity

The permanent tendency to include more and more aspects

- ◆ New kind of applications (e.g. web based)
- ◆ New kind of issues (e.g. deployment, virtual enterprise)
- ◆ New kind of requirements (e.g. dyn. Reconfigurable appl)
- ◆ + Scalability and more performance

SCM systems have a tendency to become too fat.

An SCM system is a monolithic and proprietary system.

⇒ difficulties for interoperability, incremental
introduction, transitions, evolution, cost

20



Perspectives

- In the 80s: Most concepts and techniques developed.
 - ◆ Version control (70s), Change set.
 - ◆ Workspace control,
 - ◆ Change control ...But ad hoc tools, no DB support, weak usability.
- Early 90s
 - ◆ Process support
 - ◆ Better interfaces, coverage
 - ◆ More professionally builtMaturation phase.

21



Perspective / Conclusion

- Late 90s.
Despite the remaining weaknesses, mature and successful.
An essential technology for successful SE projects ...
One of the few success in SE technology, but challenged
- Market (Ovum)
\$1 Billion in 98,
99: 25% Mainframe, 15-20% of workstations and 5-10% PC.

\$2 Billions in 2000
\$3.3 Billions in 2002

22