

# Coordinated Editing of Versioned Packages in the JP Programming Environment

Michael L. Van De Vanter  
The Forest Project  
Sun Microsystems Laboratories

8th International Symposium on  
System Configuration Management  
Brussels, 20-21 July 1998

# Overview

- The Forest Project & JP
- Versioning & CM in JP
- The Coordinated Editing Problem
- The PartHandler Interface
- How Coordination works
- Conclusions

# Background: The Forest Project

- JP Programming Environment
  - Scalable development model for Java programming
  - Enhance code reuse by JP-compliant environments
  - "write once, build anywhere, run anywhere"
- Orthogonal Persistence for Java (OPJ)
  - Extend Java programming model to storage
  - Make object lifetime independent of type, code, etc.
  - "write once, run anywhere" for data-centric apps.

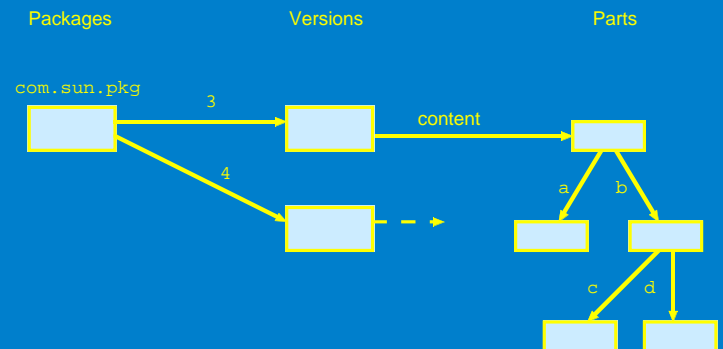
# JP: Key Technologies

- Versioned Packages (after Vesta)
  - Modular system construction
  - Storage
  - Building
  - Versioning/CM
- OO Repository
  - Java objects in OPJ
  - Strong use of interfaces

## JP: Versioning & CM

- Uniform Granularity
- Package Version Content
  - Orthogonal
  - Immutable
  - Hierarchical
  - Buildable
  - Aggregation through build imports
- Version Accretion

## Example 1: Repository Components



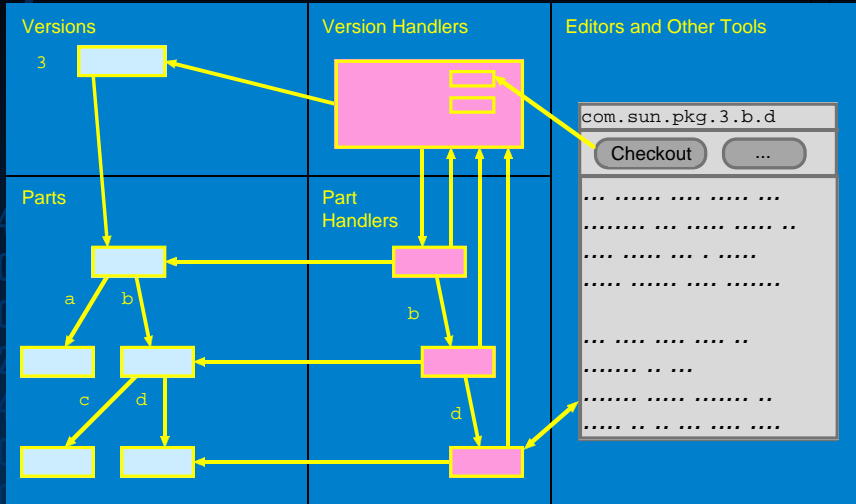
## The Coordinated Editing Problem

- How to create new versions:
  - quickly and unobtrusively
  - in a heterogeneous hierarchy of parts
  - using type-specific editors
  - collaborating through generic interfaces

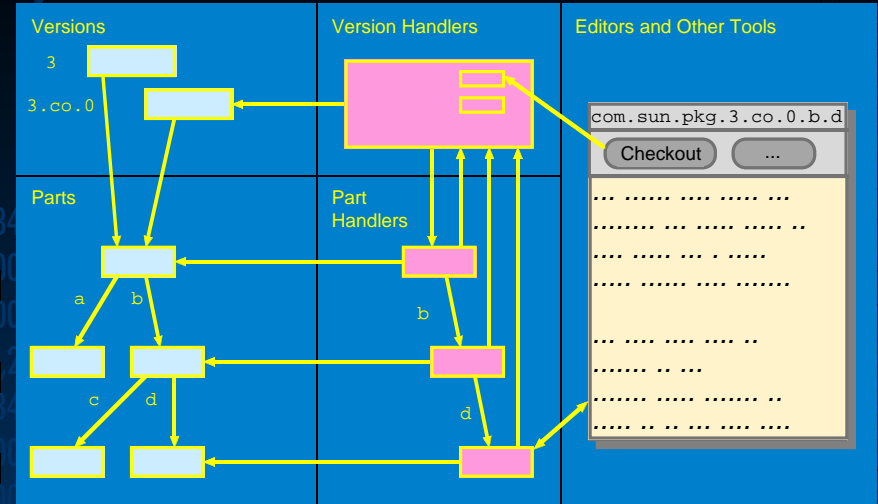
## The Key Interface

```
public interface PartHandler {  
    // Context and Coordination methods  
    void initialize(VersionHandler vh);  
    boolean setPart(Mutability part);  
    void edit();  
    PartHandler getPartHandler(Path name);  
    // Versioning methods  
    Mutability advance();  
    void revert();  
    // Usability methods  
    void setEditable(boolean isEditable);  
    void setModified(boolean isModified);  
    void setVersionName(String name);  
    void setPartName(String name);  
}
```

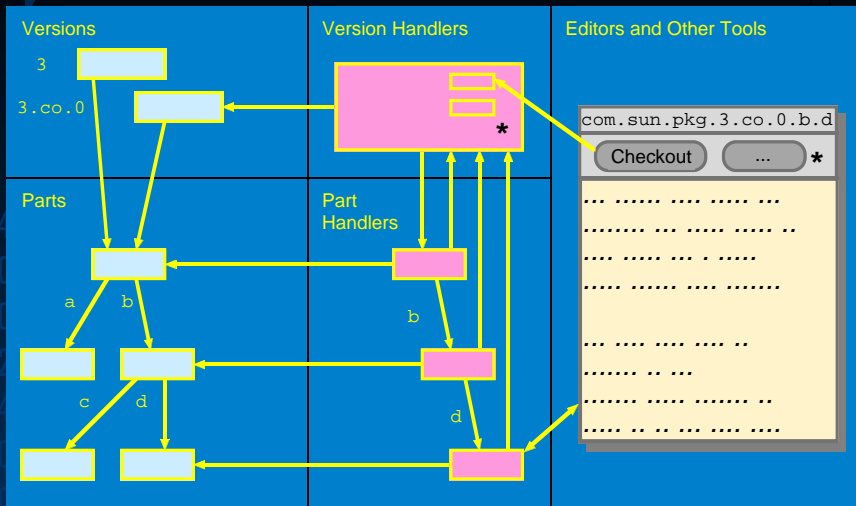
## Example 2: Handler Relationships



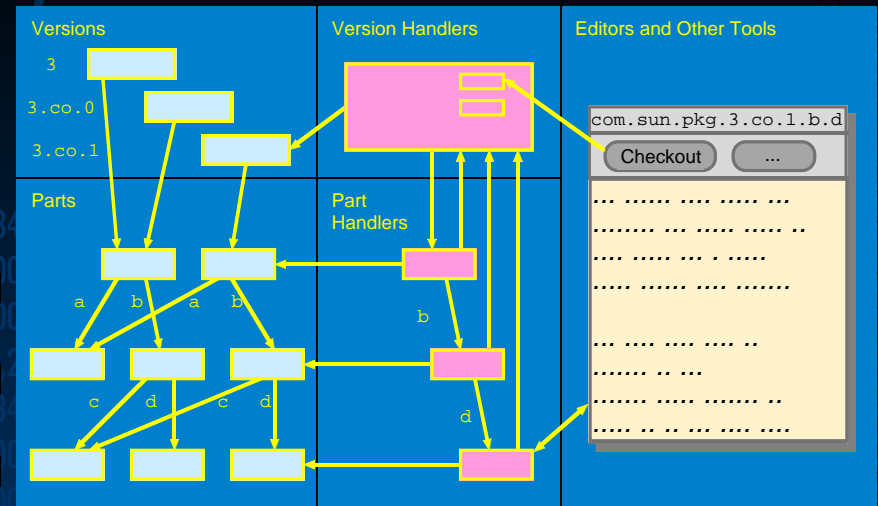
## Example 3: Versioning-Checkout



## Example 4: Editing



## Example 5: Versioning-Advance



## Conclusions

- A simple, build-oriented CM system
- Simplicity through Orthogonality
  - versioning
  - storage
- Translates to a simple user model

1,234.00  
42,009.45  
96,000,000  
100,230.00  
1,234.00  
42,009.45  
96,000,000  
100,230.00

1995-200?



Coordinated Editing of Versioned Documents in JP

## Related Work

- Vesta
- Poem
- Compound Document Editing
- COOP/Orm
- ClearCase

1,234.00  
42,009.45  
96,000,000  
100,230.00  
1,234.00  
42,009.45  
96,000,000  
100,230.00

1995-200?



Coordinated Editing of Versioned Documents in JP

## Project Status

- Implemented in JP predecessor
  - Limited experience
  - Integrated with two editors
  - Based on C++/ObjectStore
- Being reimplemented in OPJ-based JP
  - Pure Java implementation
  - All tools run in OPJ "store"

1,234.00  
42,009.45  
96,000,000  
100,230.00  
1,234.00  
42,009.45  
96,000,000  
100,230.00

1995-200?



Coordinated Editing of Versioned Documents in JP

## Current Goals

- Complete JP in OPJ implementation
- Represent sources at finer granularity
- Language-based editing
- Add process-oriented information

1,234.00  
42,009.45  
96,000,000  
100,230.00  
1,234.00  
42,009.45  
96,000,000  
100,230.00

1995-200?



Coordinated Editing of Versioned Documents in JP