

# Comparison and Versioning of Scientific Workflow

Eduardo Ogasawara<sup>1</sup>

Pablo Rangel<sup>1</sup>

Leonardo Murta<sup>2</sup>

Cláudia Werner<sup>1</sup>

Marta Mattoso<sup>1</sup>

<sup>1</sup>Federal University of Rio de Janeiro

<sup>2</sup>Fluminense Federal University



# Summary

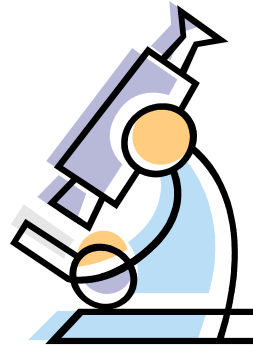
- Scientific Workflows
- Versioning of Scientific Workflows
- Diff/Merge of Scientific Workflows
- Conclusion

# 1

## Scientific Workflows and Scientific Workflows Management Systems

# Experiment Scenario

## Laboratory

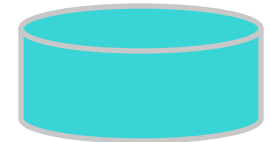
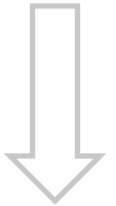


1. In vitro experiment

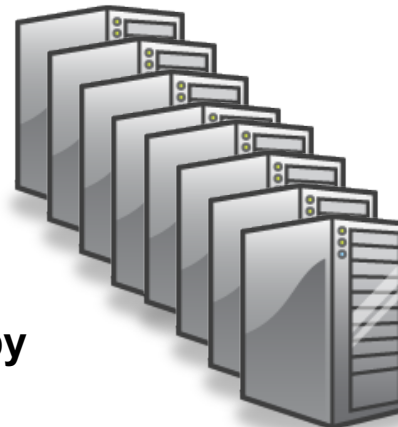
2. Data analyzed  
by program X



3. Large Volume of  
Data Produced ...



4. ...which need  
to be processed  
by program Y  
in a cluster



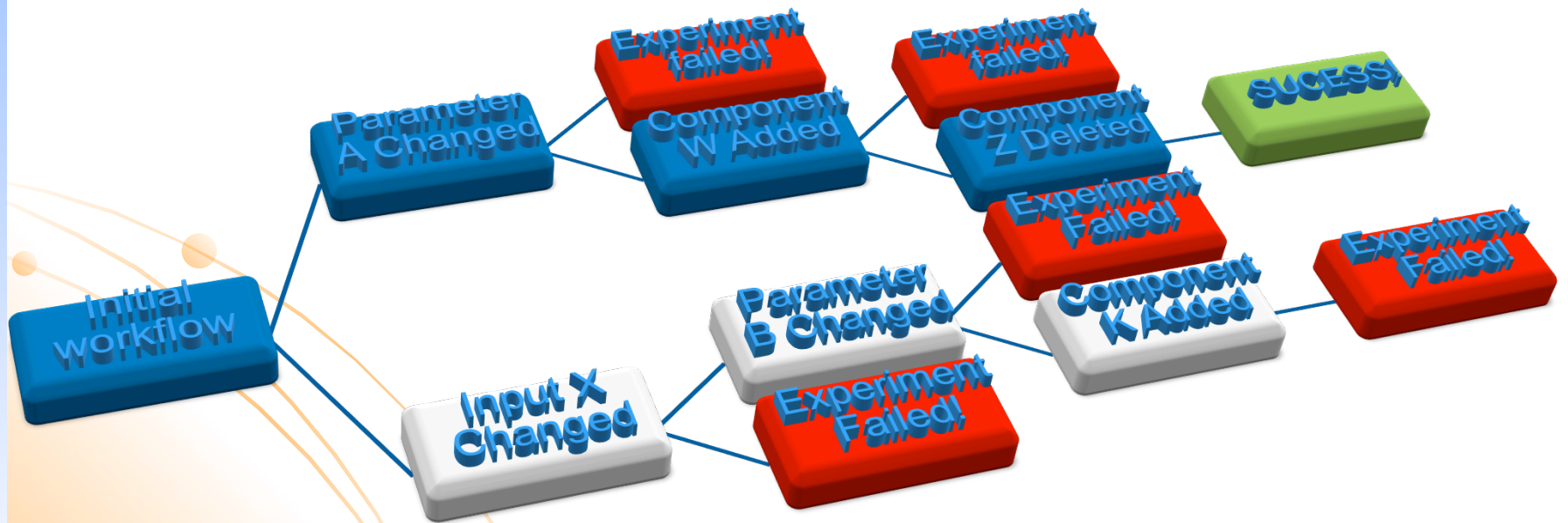
5. Results are analyzed  
by program Z



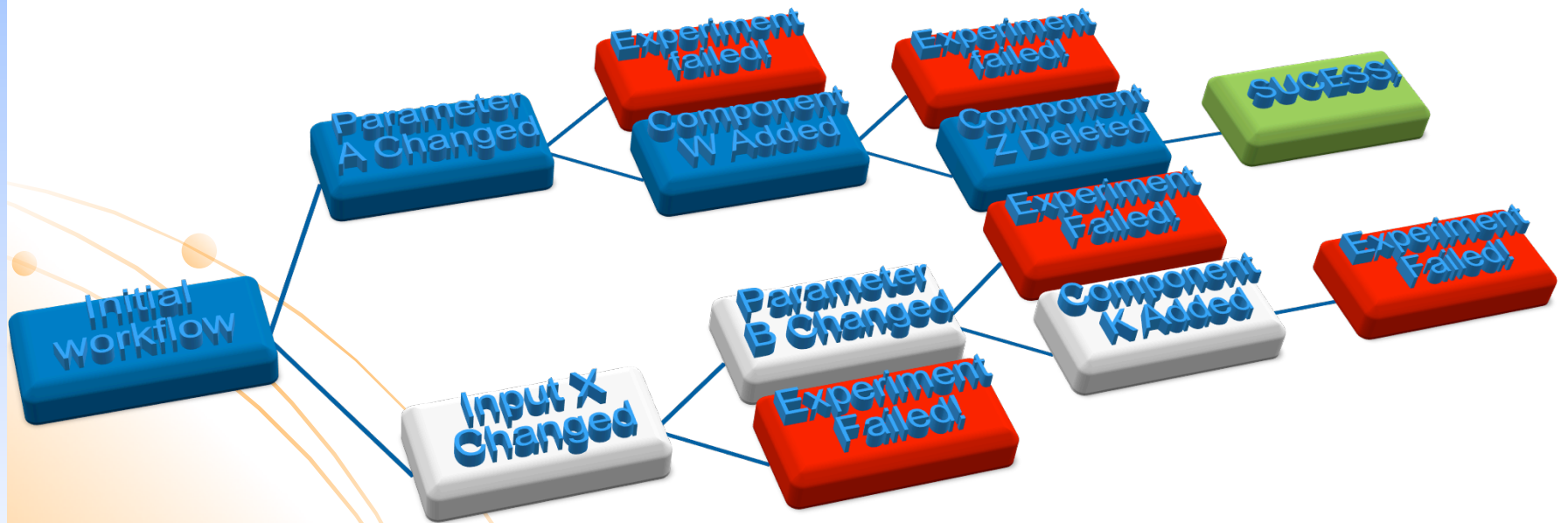
In silico experiments assisted by  
scientific workflows



# In Silico Experiment Process

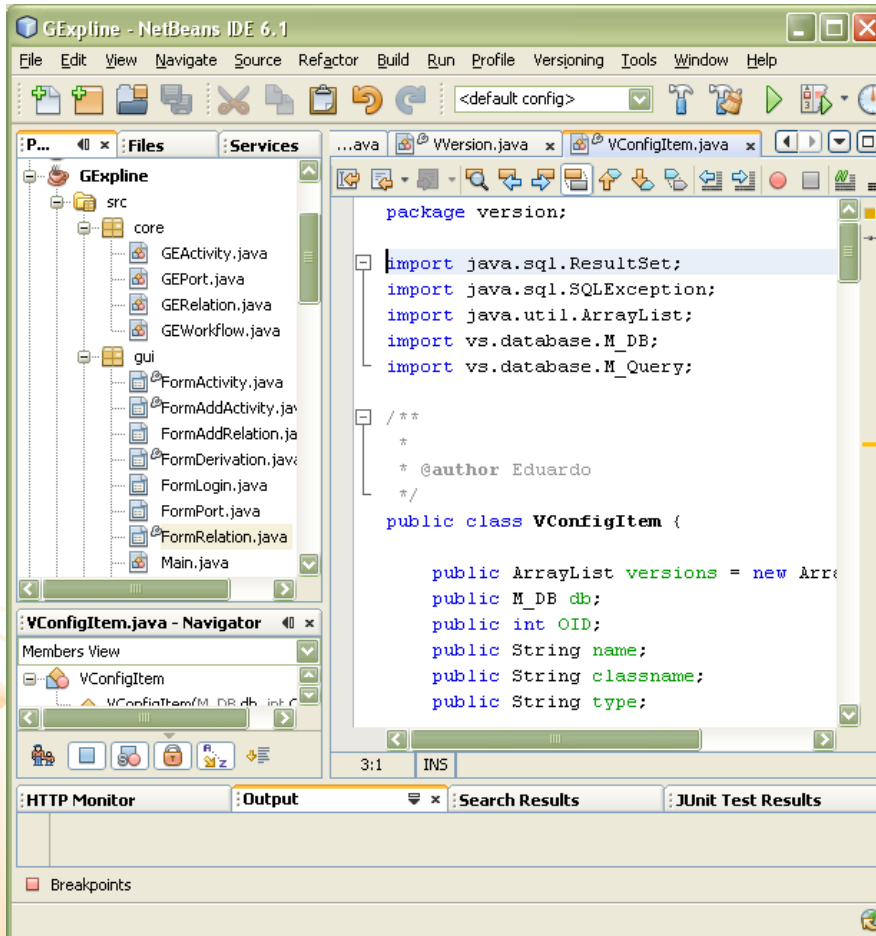


# Sharing and collaborating scientific workflows



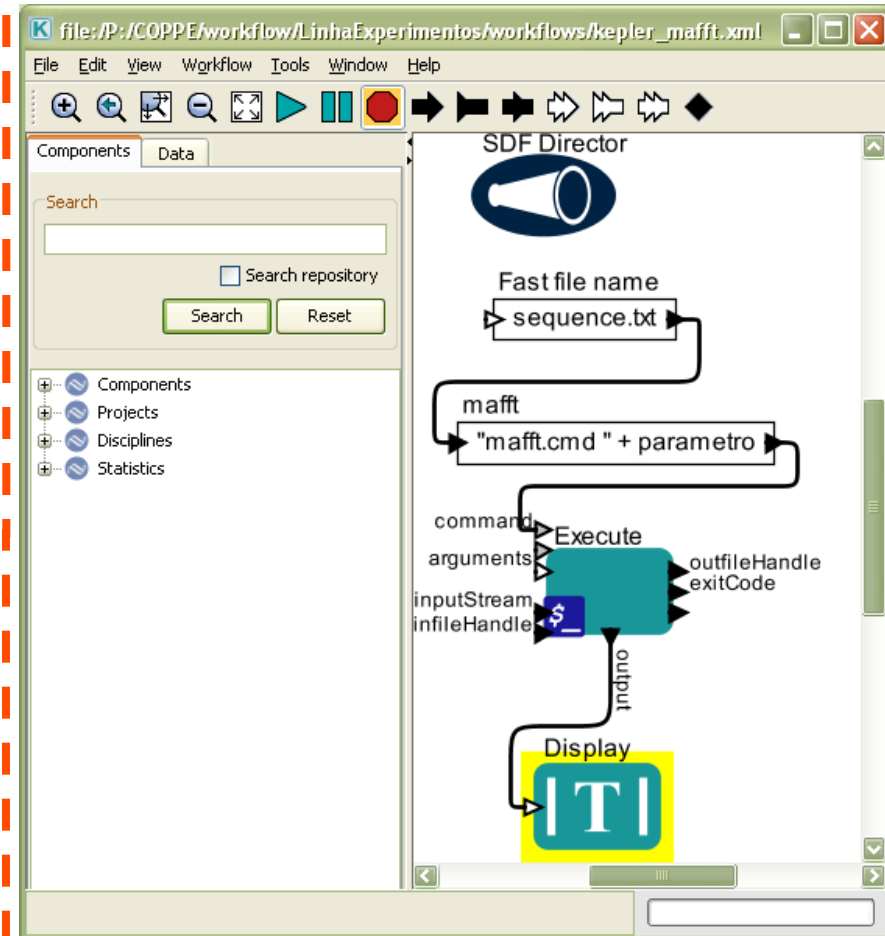
# Comparison with software development

## Programmer's IDE



**Version control system with a repository that includes diff/merge facilities for collaborative software development**

## E-Scientist's IDE



**Absence of repository offering adequate version control and diff/merge infra-structure**

# Goals

- Define a version model for scientific workflows
- Define a diff/merge strategy for scientific workflows





# 2

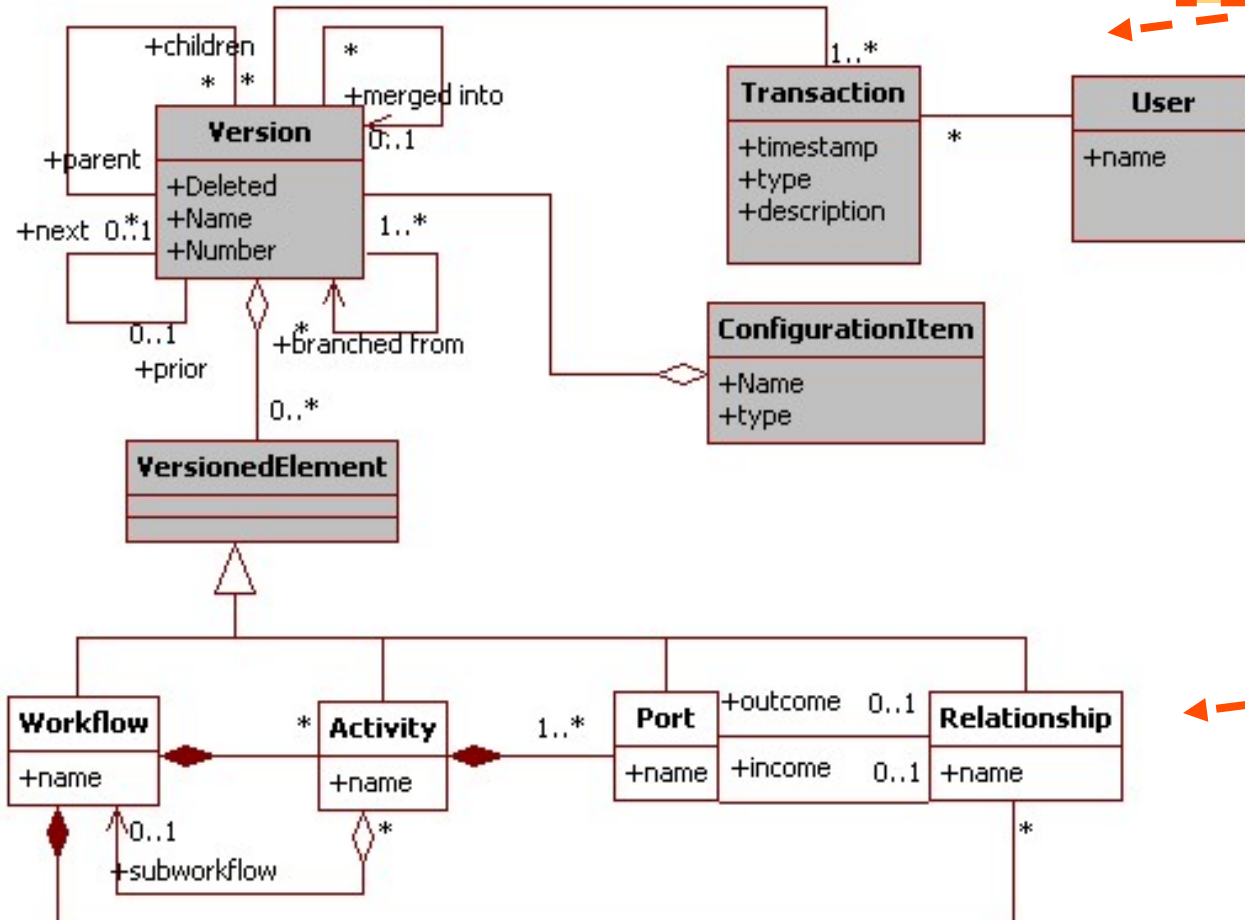
## Versioning of Scientific Workflows

# Versioning of scientific workflows

- Software process can be compared to software (Fusaro et. al., 1998) → workflows can be compared to software
- CM for workflows demands:
  - Repository with access control to register workflows and separate stable from under development versions
  - Mechanism to represent and store versions
  - Presence of workspace concept to support the modeling and maintenance of workflows

# Version model

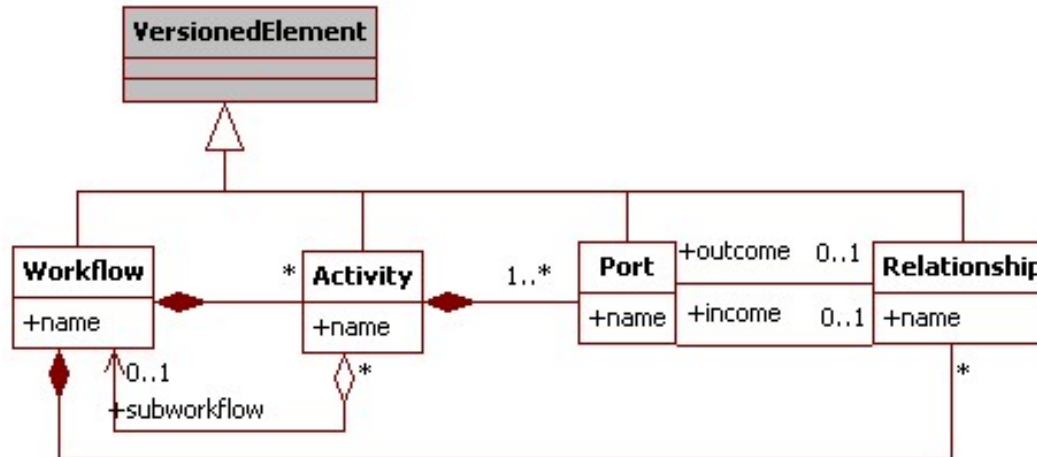
Version space



Product space

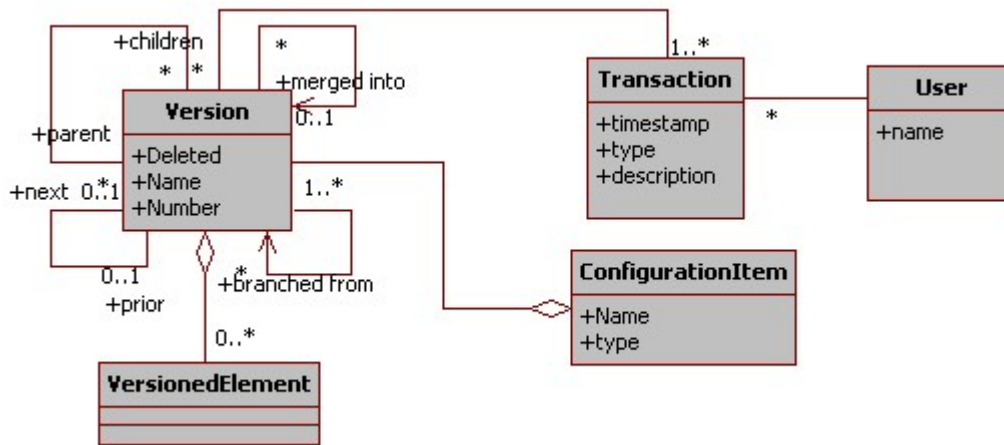
Objects to be versioned and version identification (Conradi, 1998)

# Product space



- Coarse-grained units
  - Structural information of the workflow. The graph decomposition of the workflow
  - Inherit “VersionedElement”
  - Workflow, Activity, Relationship, Ports
- Fine-grained units
  - Internal information of each class

# Version space

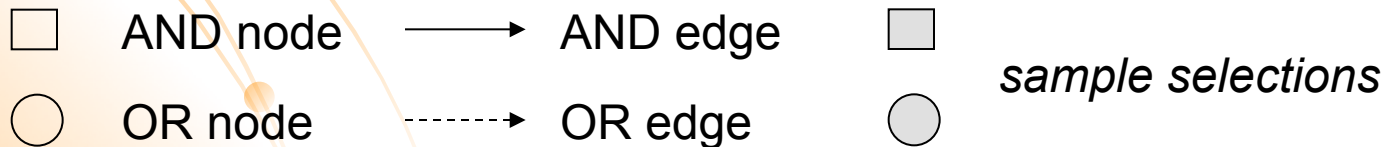
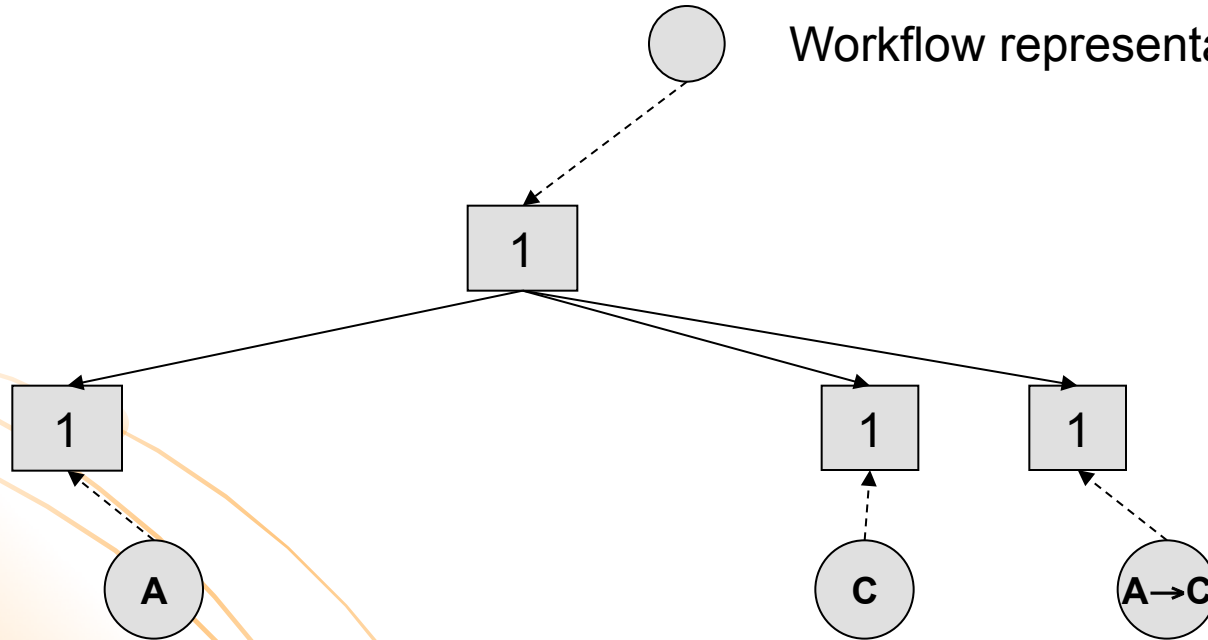


- Each ConfigurationItem is composed by Version
- Each VersionedElement has a version identifier
- Version have next\* and previous version
- A version may have branches and may be merged

# Interplay between version and product space

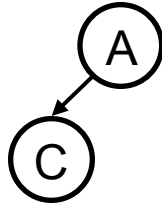


Workflow representation in version space



# Workflow evolution

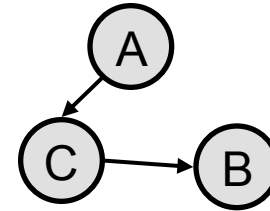
Version 1



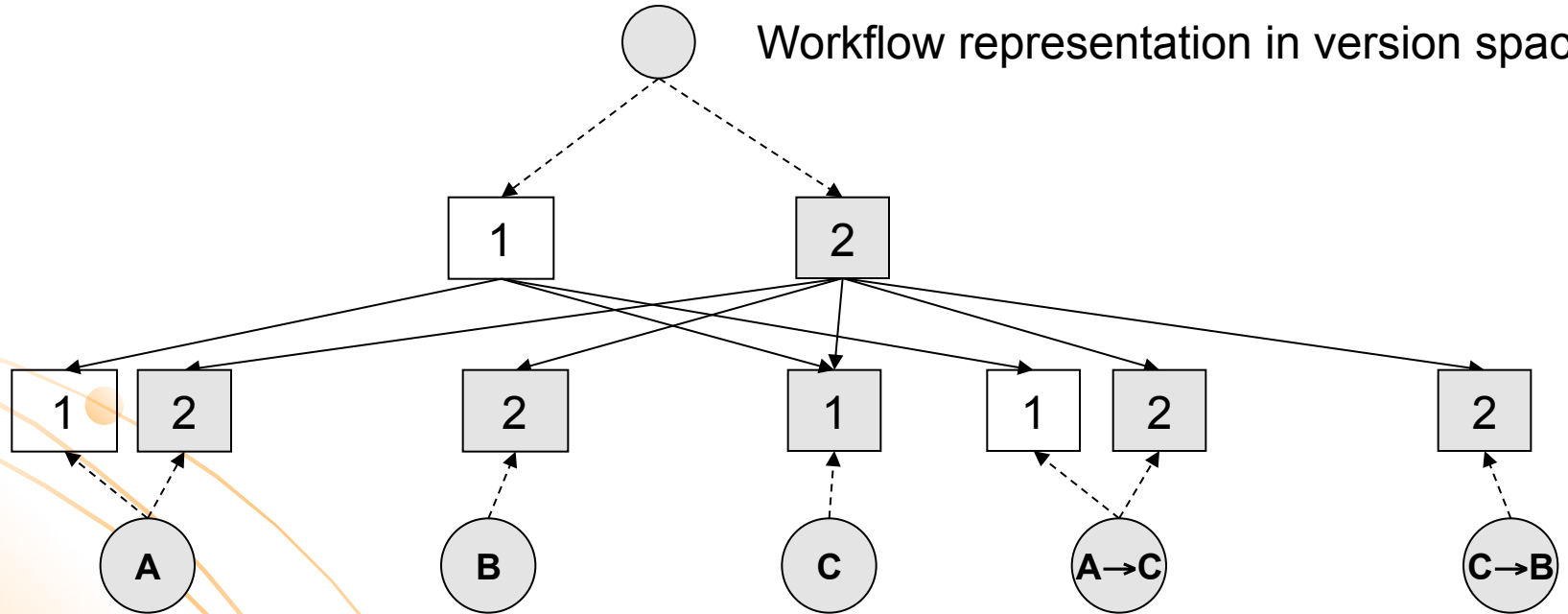
Workflow structure



Version 2



Workflow representation in version space



□ AND node

○ OR node

→ AND edge

- - - - - OR edge



*sample selections*

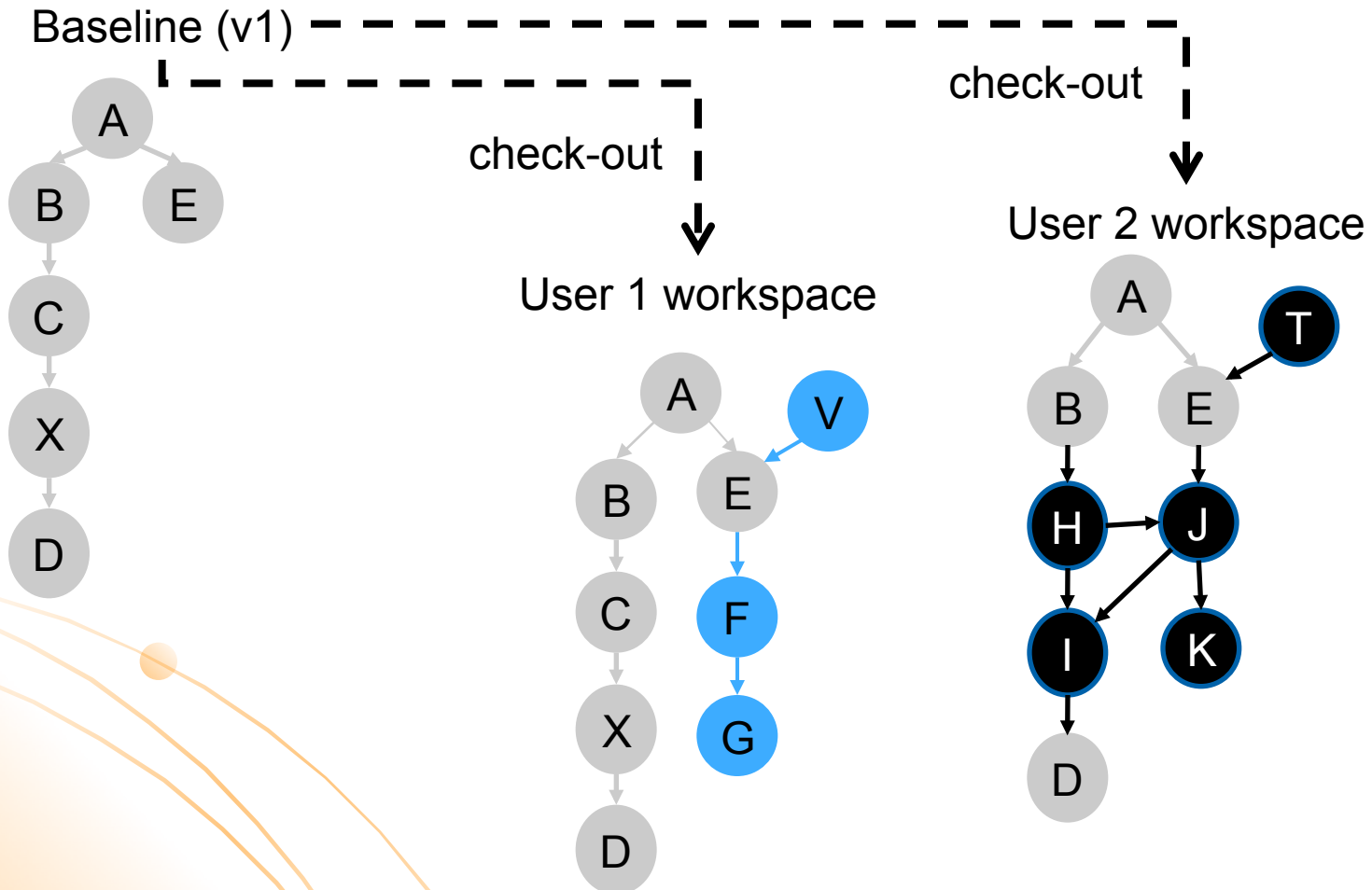


# 3

## Diff / Merge of Scientific Workflows

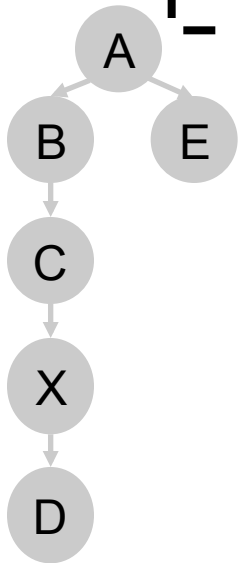


# collaborative scenario



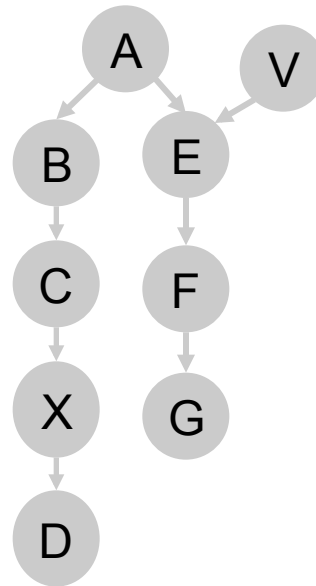
# User 1 makes check-in

Baseline (v1)

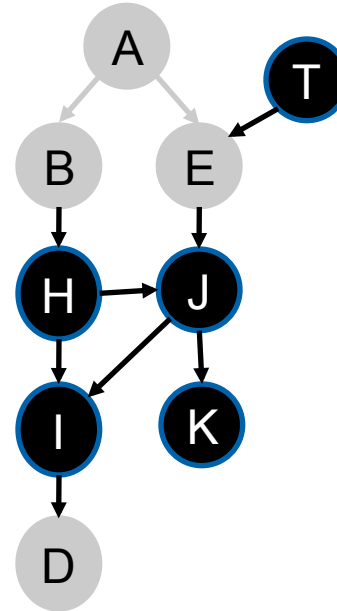


check-out

Current version (v2)

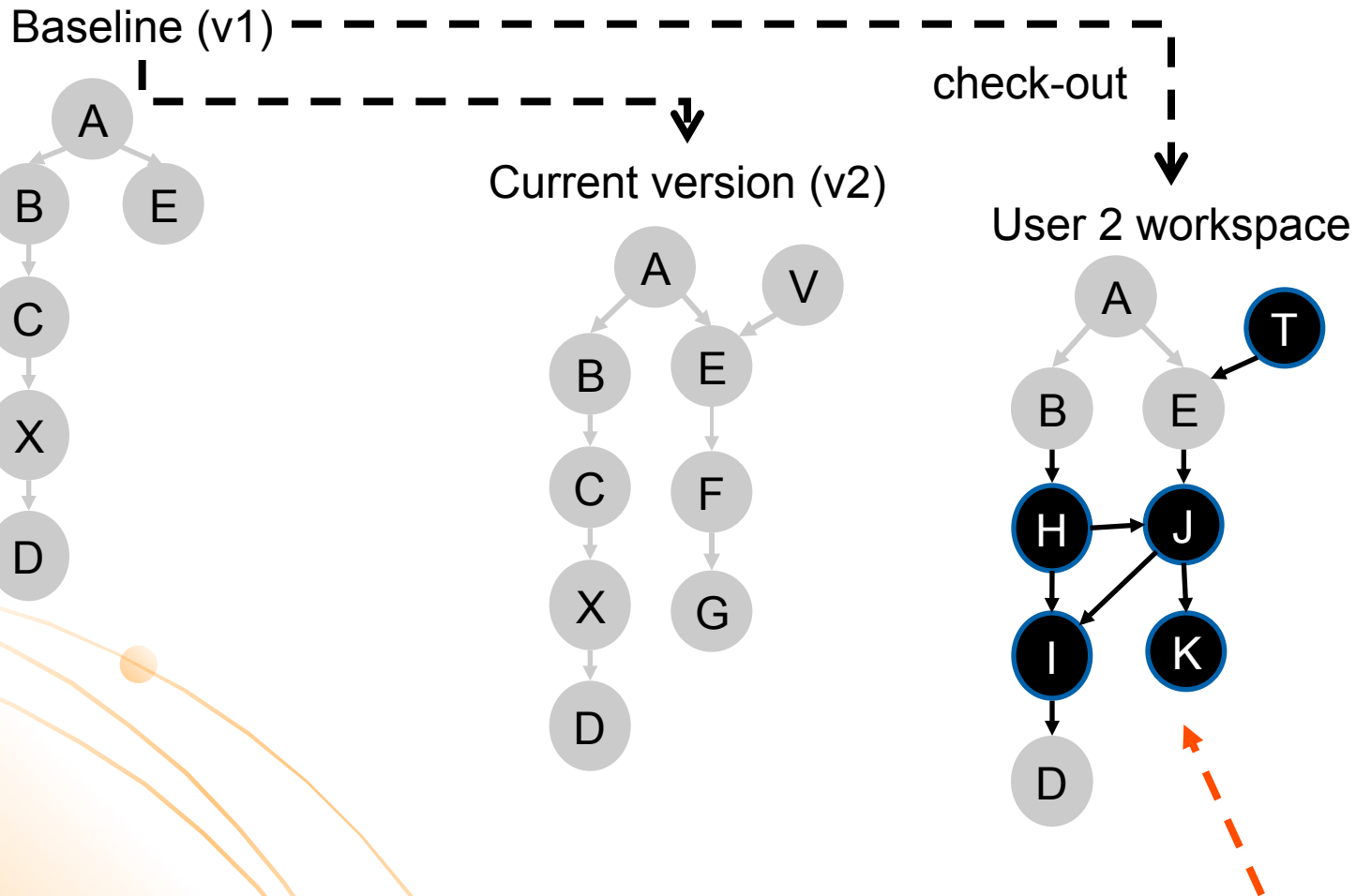


User 2 workspace



User 1 tries to check-in first.  
No problem.

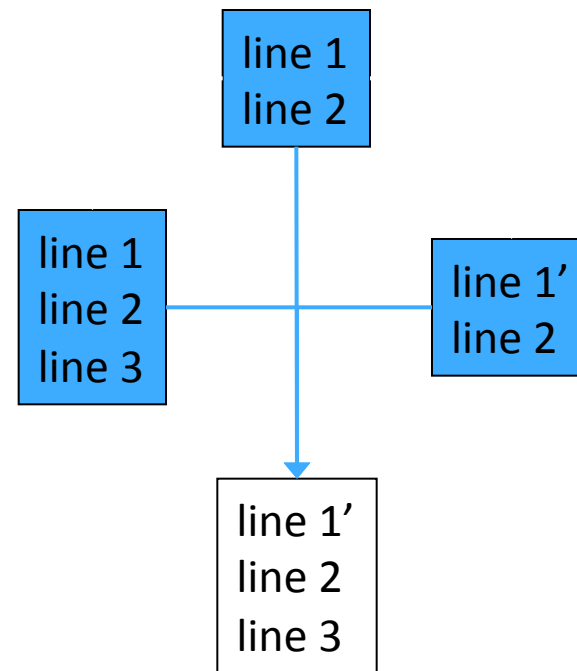
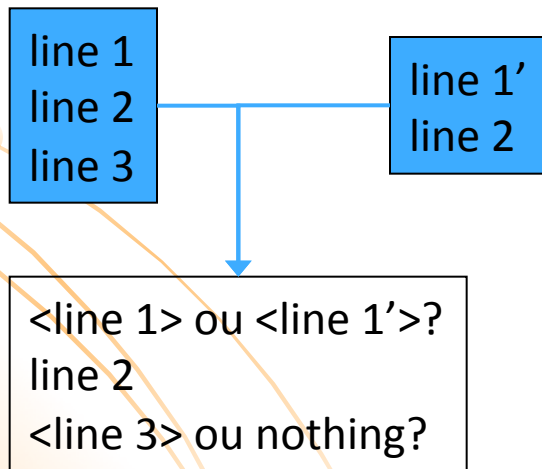
# User 2 tries to check-in



**When User 2 tries to check-in, a diff/merge needs to be executed.**

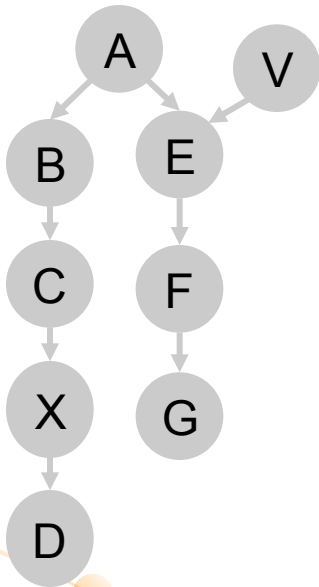
# diff / merge

- Configuration management tools usually supports:
  - *2-way merge*
  - *3-way merge*

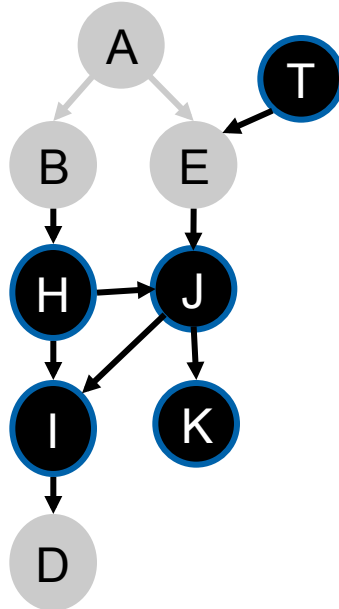


# 2 way merge

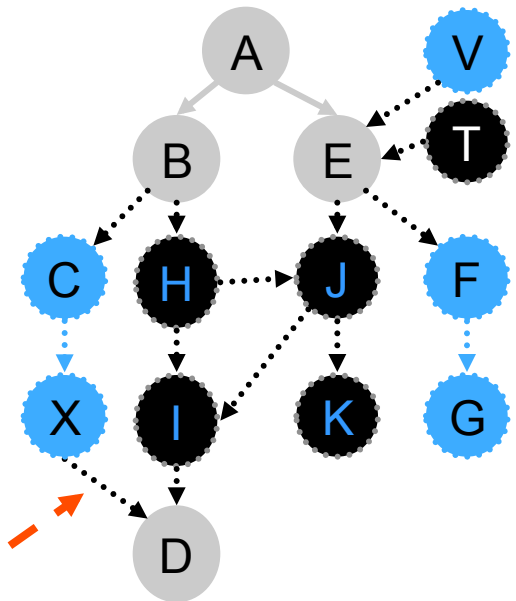
Current version (v2)



User 2 workspace

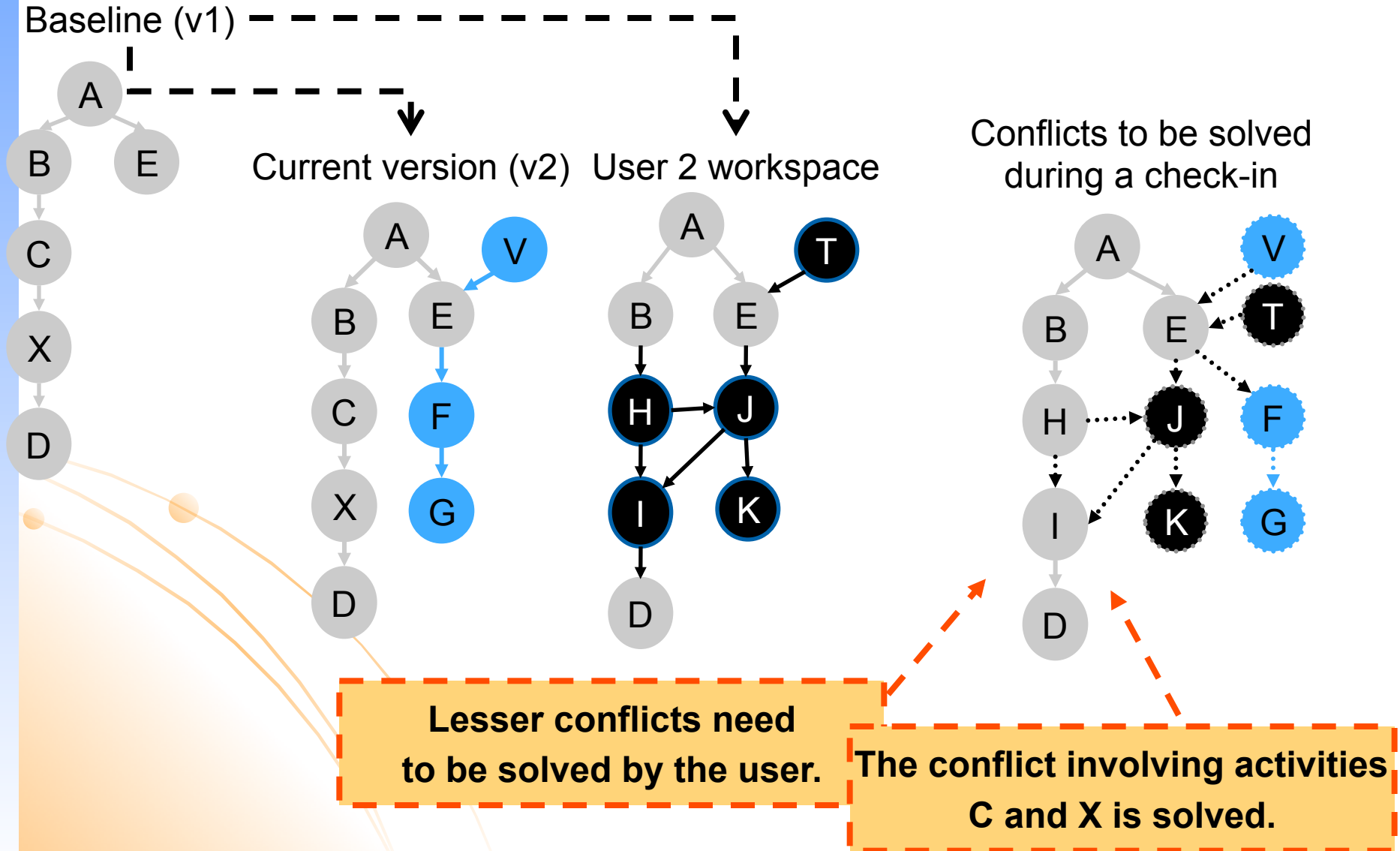


Conflicts to be solved during a check-in



**All conflicts need to be solved by the user.**

# 3 way merge

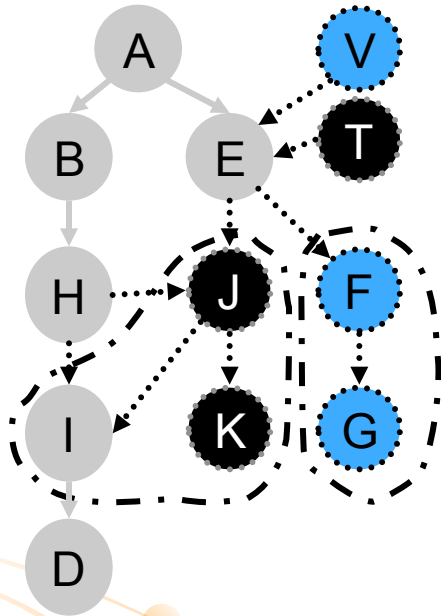


# 3-way sub graph diff/merge

- Workflows have a dual behavior of being a model and executable code at the same time
- Goal is to support a syntax merge, which means that a candidate conflict is not just a coarse grain unit, but a sub graph from the initial coarse grain conflict unit

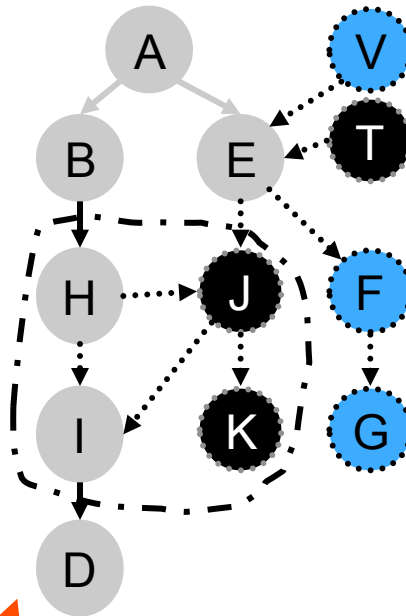
# 3-way sub graph diff/merge

step 1



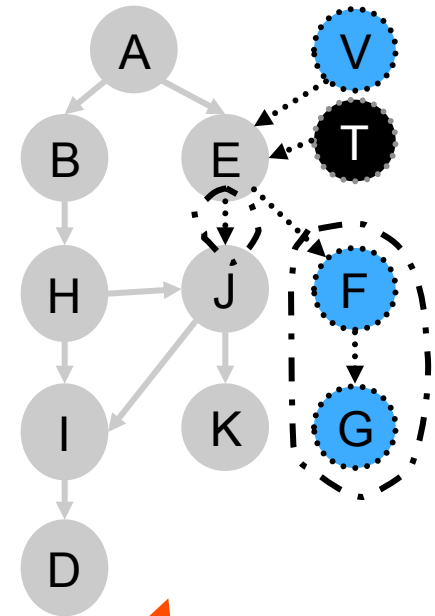
Exploring from E:  
No simplification

step 2



Exploring from B:  
Simplification by  
syntax merge

step 3

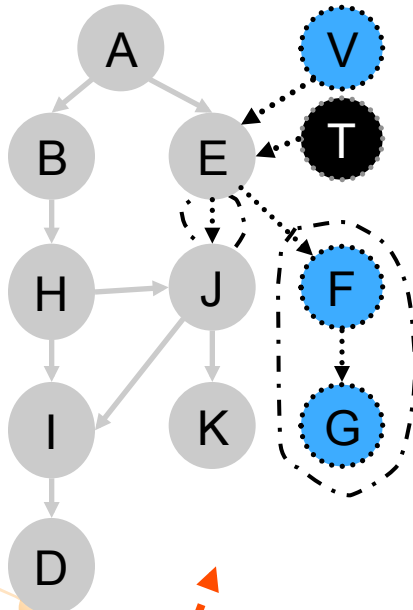


Re-exploring from E:  
decrease in conflict size



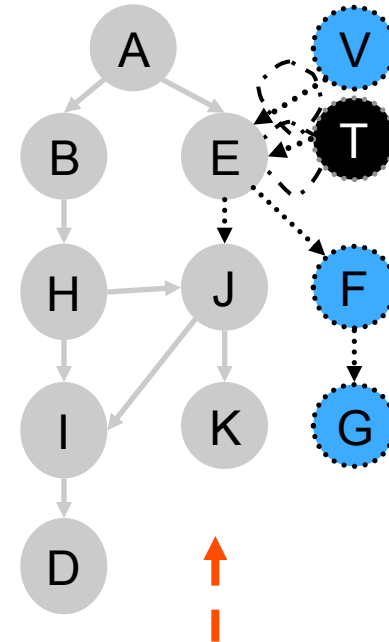
# 3-way sub graph diff/merge - two final conflicts

Conflict one



Exploring from E  
In forward direction

Conflict two



Exploring from E  
in backward direction

# 4 Conclusions

# Conclusions

- Contributions
  - Version Model for scientific workflows
  - Syntax diff/merge for scientific workflows
- Prototype under development:
  - Evaluate presented concepts
  - Developed on top of Java Workflow Editor (<http://www.enhydra.org>) using Postgresql DBMS

# Comparison and Versioning of Scientific Workflows

# Thank you!

Visit our Web site  
<http://gexp.nacad.ufrj.br>

Eduardo Ogasawara  
[ogasawara@cos.ufrj.br](mailto:ogasawara@cos.ufrj.br)

