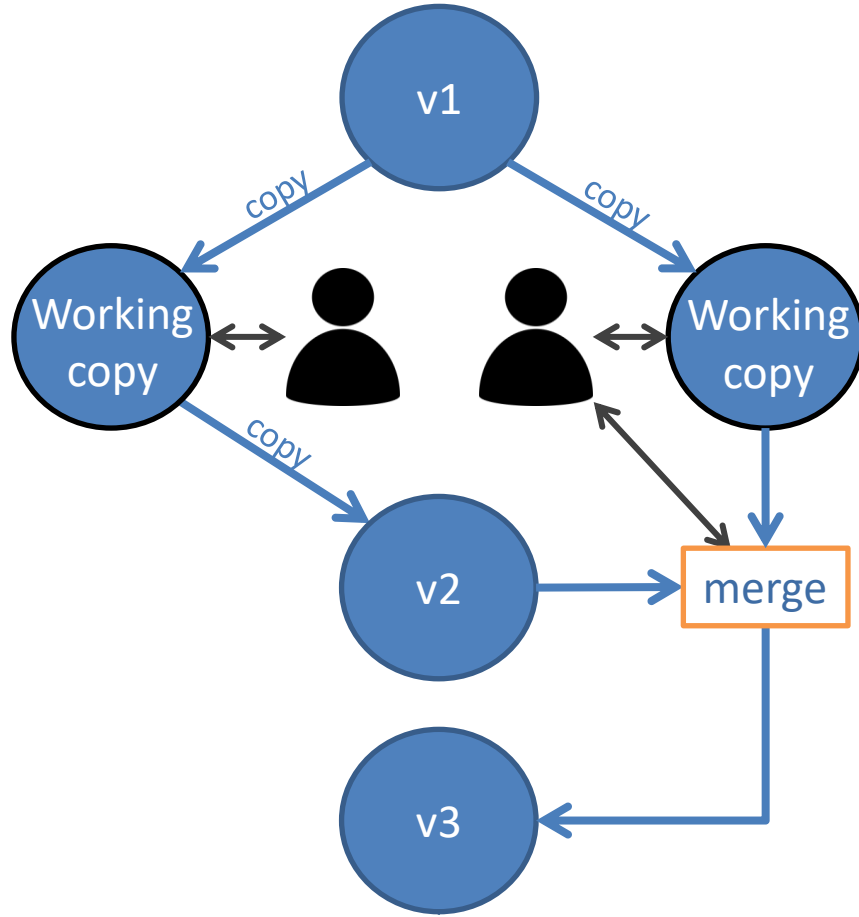


# **Collaborative Modeling with Version Control**

BigMDE, 21 July 2017  
Steven Kelly, MetaCase

# Clone

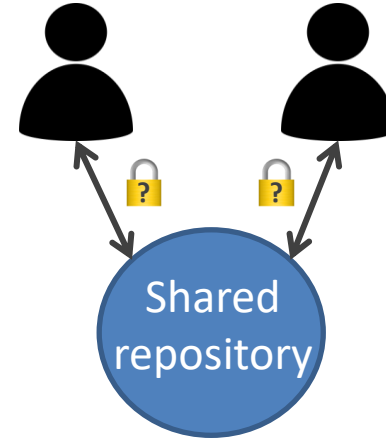
*merge after the fact*



vs.

# Share

*continuous integration*



**versions vs. multi-user**  
**flat files vs. database**  
**ASCII vs. objects**  
**text vs. graphics**

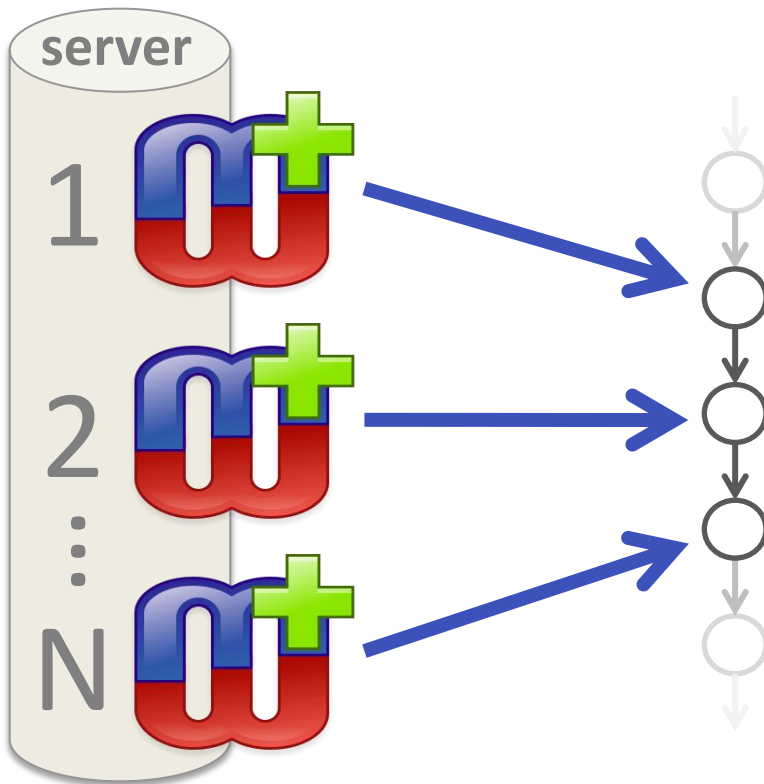
# The challenge: mix oil & water

versions ↔ multi-user

flat files ↔ database

ASCII ↔ objects

text ↔ graphics



# GitHub

# MetaEdit+

# Four purposes for versioning

- Understanding what's been done
  - Diff, version comments
- Archive and backup
- Branching
  - Parallel variants
  - Releases
- Collaboration

New tools



VCS integration



DSML

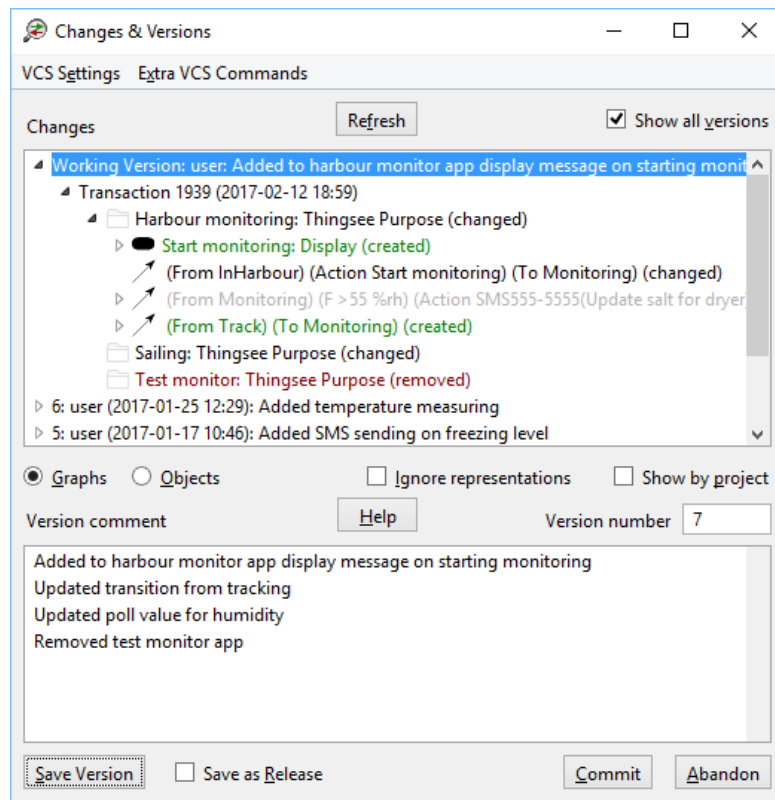


Multi-user



# Changes and Versions tools

- Automatic trace of model changes
- See changes graphically directly in your models
- View changes as a tree using your language's structure and symbols, not XML
- Compare changes as a textual diff with live links to models
- Filters (your changes/all changes, data/representation)



# Three ways to view changes

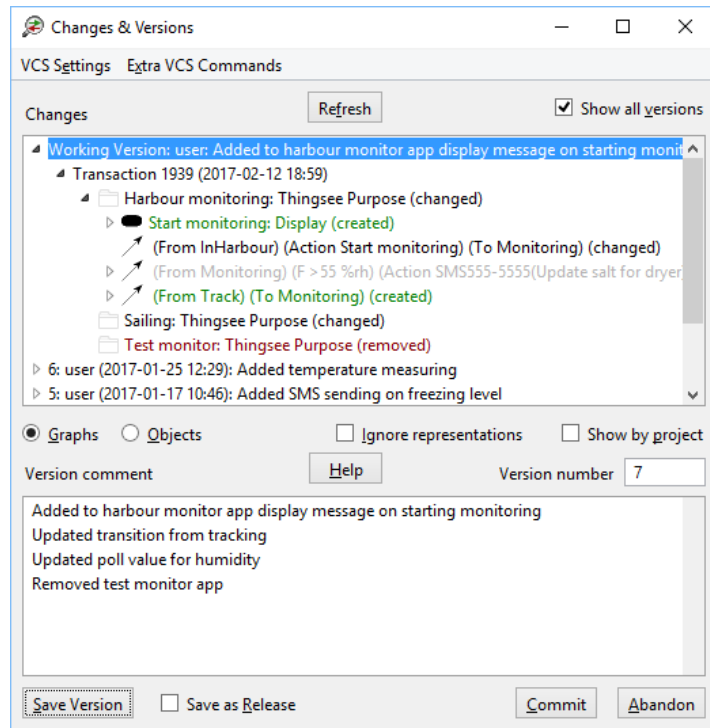
The screenshot displays the 'Changes & Versions' window, which is divided into three main sections illustrating different ways to view changes:

- Changes Panel (Left):** Shows a tree view of changes for the 'Working Version: user'. The changes include:
  - Transaction 383 (2017-01-25 15:05)
    - My apps: Thingsee Profile
      - Harbour monitoring: Thingsee purpose (changed)
        - Start monitoring: Display (created)
        - 2 lux: Luminance (changed)
        - InitializeSailing: Other Purpose (changed)
        - < 0 C: Temperature
          - (From InHarbour) (Action Start monitoring) (To Monitoring) (changed)
        - Speeding Alert: Thingsee purpose (removed)

- Comparison Panel (Bottom Left):** Shows a comparison of changes for 'Harbour monitoring: Thingsee purpose changes in Transaction 383 (2017-01-25 15:05)'. It lists various roles and objects, such as 'Start state: F', 'Always active: F', 'Description: Object: Monitoring <State> 3\_8332', and 'Role: From <From> Object: <State> 3\_8053'.
- Diagram Panel (Right):** A diagram titled 'Monitoring' showing the flow of data and actions. It includes components like 'InHarbour', 'Start monitoring', 'Bump detection', 'Impact > 1.05 (g)', '0(Warning: water freezing.)', '< 0 C', '> 55 %rh', '0(Update salt for dryer)', and 'InitializeSailing'. The diagram illustrates the relationships between these components and the changes made in the transaction.

# Versioning with external VCSs

- Simple setup and use
- Automated background execution of versioning commands
- Maintain same model in several places, syncing with GitHub etc.
- Version and diff your language and generators along with models
- Git, SVN predefined
  - extend with your own



# Version Control Integration (e.g. Git)

The image displays a GitHub commit page on the left and a Thingsee configuration window on the right, illustrating the integration of version control with IoT device configuration.

**GitHub Commit Page (Left):**

- Repository: `mccjpt / IoT`
- Commit: `5: user (2017-01-30 12:46Z)`
- Message: `Added start monitoring display, changed humidity sensor polling, removed impact`
- Files changed: 1 file
- Showing 24 changed files with 4 additions and 7 deletions.
- File: `Harbour monitoring_3_8156_Thingsee purpose.txt`
- Diff summary:
  - @@ -9,10 +9,6 @@ Object: Impact >1.5 (g) <Acceleration> 3\_8899
  - 9 9 Direction: Any
  - 10 10 Trigger if over (g): 1.5
  - 11 11 Trigger if under (g): 0
  - 12 -Object: Impact >1.05 (g) <Acceleration> 3\_8943
  - 13 - Direction: Any
  - 14 - Trigger if over (g): 1.05

**Thingsee Configuration Window (Right):**

Thingsee purpose: Harbour monitoring, 14 September 2015, 16:27

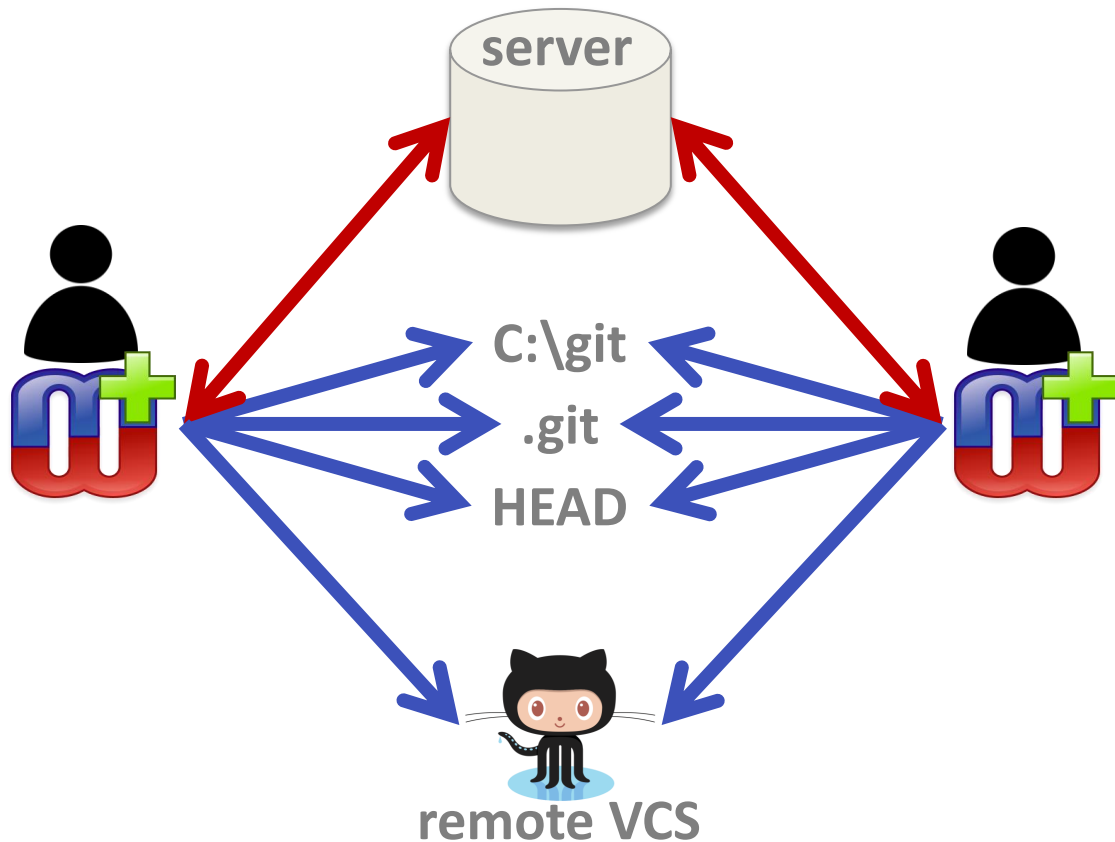
The configuration diagram shows a central **Monitoring** block connected to various sensors and actuators:

- InHarbour** sensor connected to the **Monitoring** block.
- Start monitoring** button (highlighted with a red box) connected to the **Monitoring** block.
- Bump detection** sensor connected to the **Monitoring** block.
- Impact >1.5 (g)** sensor (highlighted with a red box) connected to the **Monitoring** block.
- Temperature <2 C** sensor connected to the **Monitoring** block.
- Humidity >55 %rh** sensor connected to the **Monitoring** block.
- 0(Warning: water freezing.)** actuator connected to the **Monitoring** block.
- 0(Update salt for dryer)** actuator connected to the **Monitoring** block.

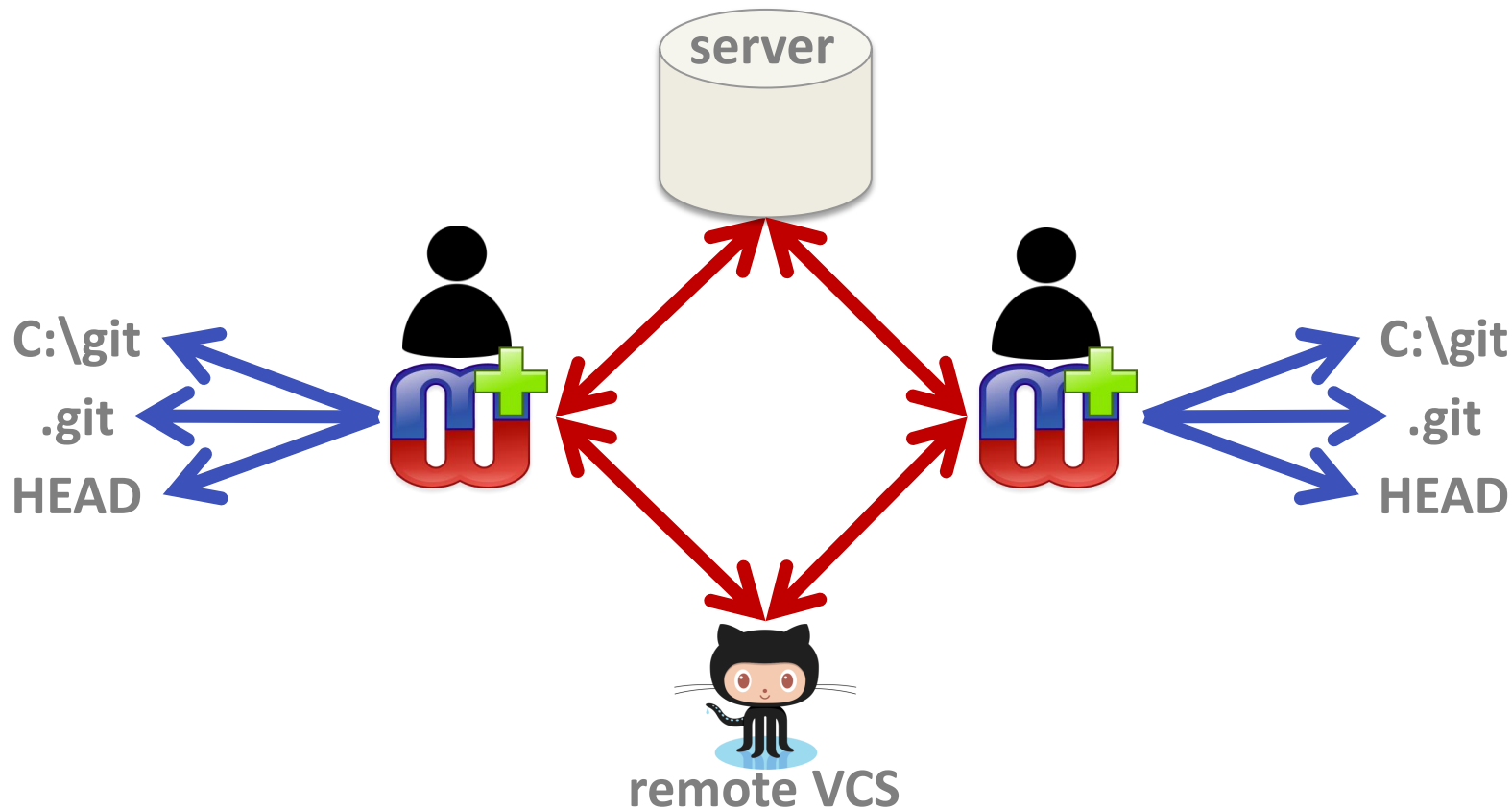
The bottom status bar indicates: **Active: Selection of 13 elemen** and **Grid: 10 @ 10**.



# Shared repo, shared working dir?



# Shared repo, working dir per user



# Persuading VCSs to leave well alone

- MetaEdit+ has already integrated all users' work
- However, local working VCS info may be behind remote
  - Another user may have versioned after we last versioned
  - VCS will try to merge, reverting files to previous user's ☹
- Need to get VCS up to date, without changing files
  - Sync with remote
  - Update local to remote HEAD
  - Accept exactly these files as next version
- Order of these and actually writing files differs per VCS:
  - Git: `write files`, remote update, `reset --soft`, `add -a`, `commit`, `push`
  - SVN: `svn update`, `write files`, TortoiseSVN `commit`

# Conclusion

- Extended a multi-user modeling tool with versioning
- New native tools for diff & working with versions
  - Three different integrated UIs: tree, graphical, (hyper)text
- Built integration with external VCSs
  - Easy, consistent UI across all VCSs, hides complexity of Git
  - Order of magnitude faster & simpler than best in Eclipse
  - User-extendable to new VCSs
- Works with multi-user repository: best of both worlds
  - No merges, no lock-outs
- Further work: for rare offline cases, add merging import

**Thank you!**