

Implementierung einer Unterstützung von Versionsverwaltung in einem Plug-In für verteilte Paarprogrammierung

Gliederung

- Verteilte Paarprogrammierung mit Saros
- Versionsverwaltung mit Git
- Unterstützung von Git in Saros
 - Begründung
 - Anforderungen
 - Implementierung
 - Zusammenfassung
- Prototyp

Verteilte Paarprogrammierung

mit Saros

Anforderungen:

- Gemeinsames bearbeiten von Artefakten
- Kommunikation (Chat und verfolgen)

Plug-In für die Entwicklungsumgebung Eclipse
Open Source

Versionsverwaltung mit Git



Anforderungen:

- Speichern von Dateien
- Verwalten von mehreren Versionen (bei Git: Commits)
- Koordination von verschiedenen Entwicklungen

Effektiv in Speichernutzung

Verteiltes Arbeiten

Begründung für Unterstützung von Git in Saros

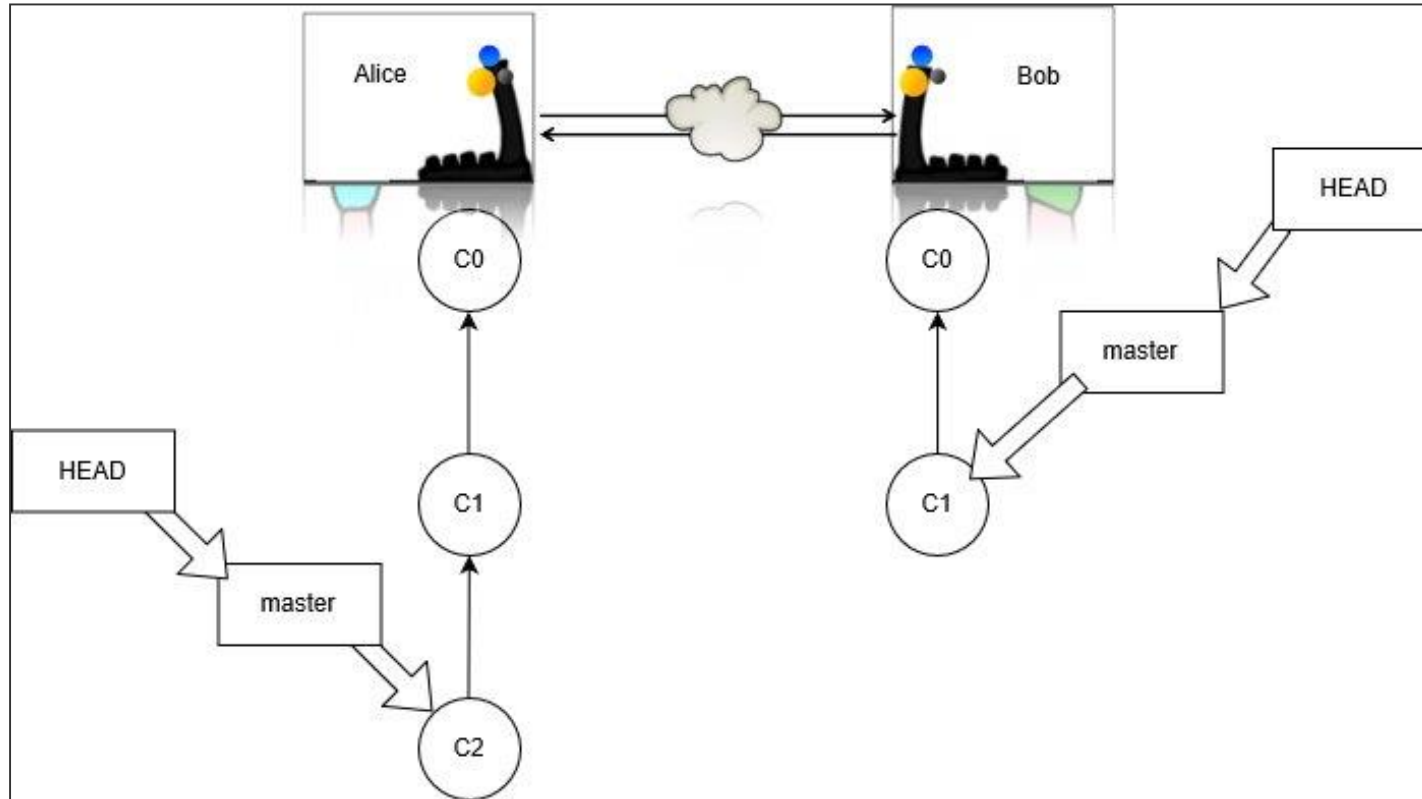
- Versionsverwaltung ist weit verbreitet
Aus den Antworten resultiert, dass alle Teilnehmer Versionsverwaltungssysteme einsetzen
(aus einer Umfrage in [1])
- Git hat genannte Merkmale und ist weit verbreitet
- Anforderungserhebung hat Wunsch bestätigt

Anforderungen - Erhebung

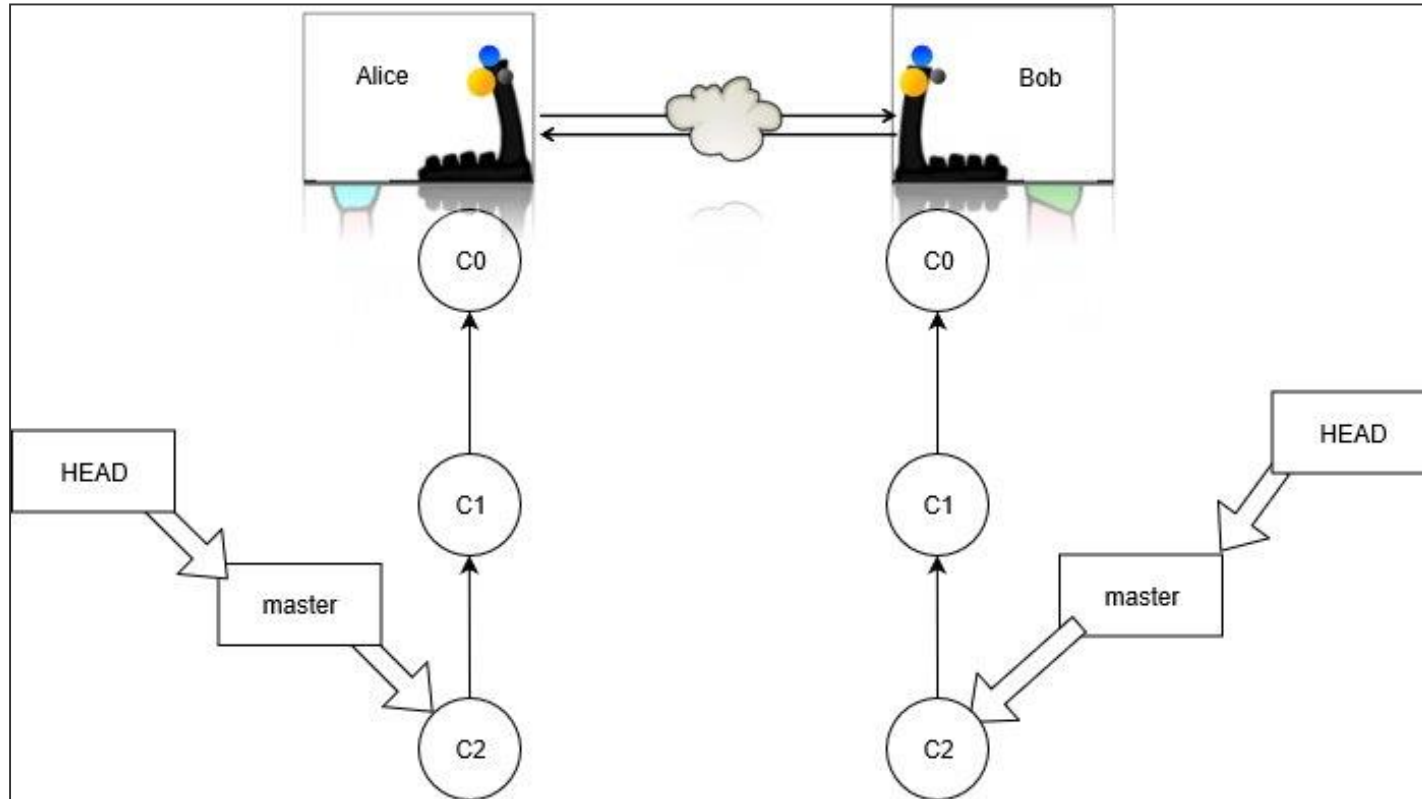
Ergebnisse:

- Entwickler besitzen gleiches Projekt mit kleinen Unterschieden
- Versenden von **einzelnen** Commits soll unterstützt werden

Anforderungen – Szenario



Anforderungen – Szenario



Anforderungen - Ziele

Grundfunktionalität:

- Versenden von einzelnen Commits beim Zusammenarbeiten
- Dokumentation
- Vermeidung von Datenverlust

Erweiterte Funktionalität

- Szenario erweitern / Abschwächung der Annahmen
- Automatisch Annahmen überprüfen
- Unit- und Integrationstests für die eigene Implementierung
- GUI

Implementierung

Saros wird größtenteils in Java entwickelt

Unterstützung in JGit

- Java-Bibliothek für die Verwendung von Git
- Aktuell
- Open Source

Implementierung – JGitFacade

Erstellen von Bundles

Fetch von Bundle

Merge

Implementierung – Funktionen der JGitFacade

git bundle

- Verwendung: Übertragung ohne Git Protokoll
- Erstellt komprimierte Datei mit mehreren Commits (Bundle)

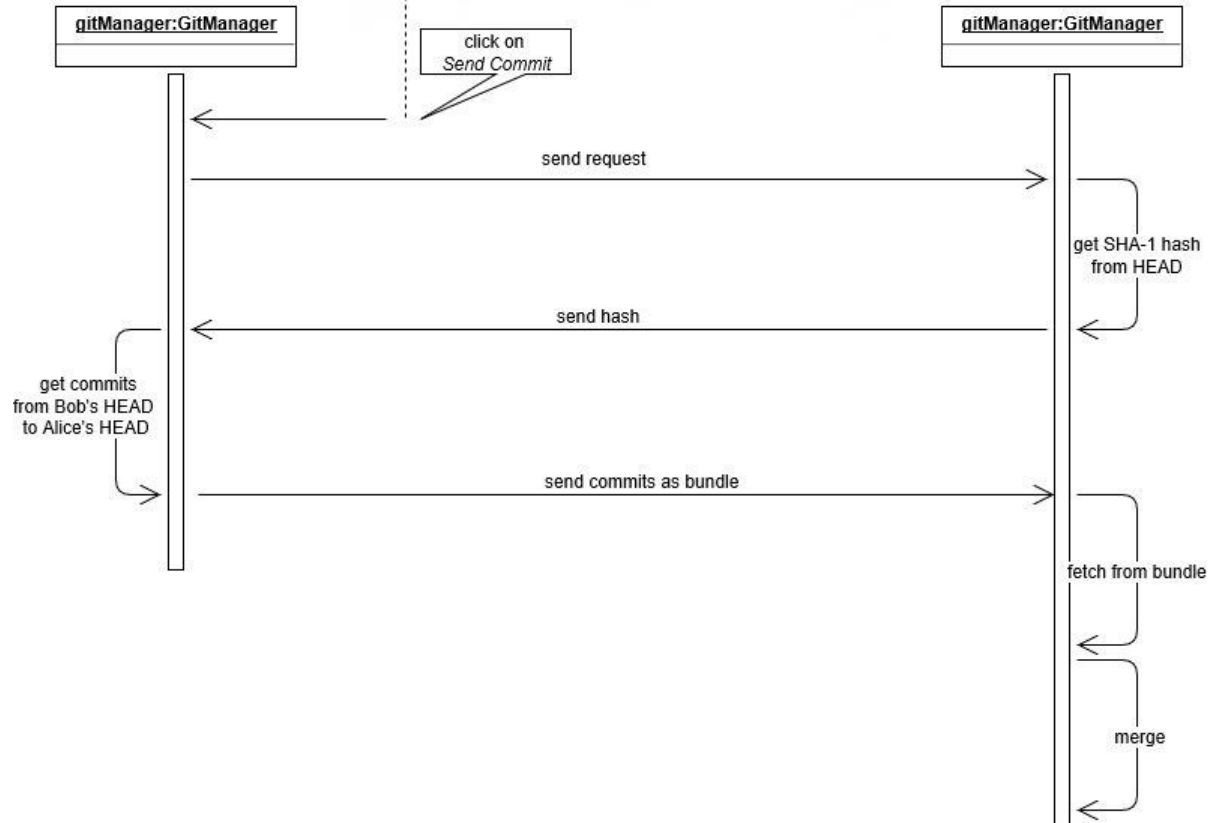
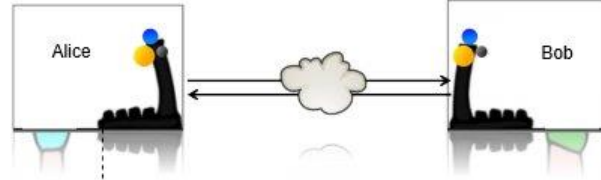
git fetch

- Verwendung: Aktualisierung
- Verbindung zwischen eigener Versionverwaltung und
 - einer weiteren Versionsverwaltung oder
 - einem Bundle

Implementierung – Funktionen der JGitFacade

git merge

- Verwendung: Zusammenführen
- Einschränkungen für Nutzung mit Saros:
 - Fast-Forward-Merge
- Annahme für alle Dateien die von Saros geteilt und in Versionsverwaltung:
 - Bei Empfänger keine Dateiänderung die nur in Zwischenspeicher von Versionsverwaltung
 - Bei Sender keine Dateiänderung im Arbeitsverzeichnis die nicht in einem Commit, aber Zwischenspeicher beeinflusst Ergebnis nicht



Zusammenfassung

Grundfunktionalität:

- Versenden von einzelnen Commits beim Zusammenarbeiten
- Dokumentation
- Vermeidung von Datenverlust

Erweiterte Funktionalität

- Szenario erweitern / Abschwächung der Annahmen
- Automatisch Annahmen überprüfen
- Unit- und Integrationstests für die eigene Implementierung
- GUI

Danke für ihre Aufmerksamkeit

Referenzen

[1] Riad Djemili. *Entwicklung einer Eclipse-Erweiterung zur Realisierung und Protokollierung verteilter Paarprogrammierung*. Freie Universität Berlin. 2006

Grafiken (Links besucht am 20.03.2019):

- Seite 3. Saros Logo. <https://raw.githubusercontent.com/saros-project/saros/master/docs/assets/images/logo/saros-logo-100x100.png>
- Seite 4. Git Logo. <https://git-scm.com/downloads/logos>
- Seite 7,8,14. Selbst erstellte Diagramme bearbeitet von <http://saros-project.github.io/saros/> mit <http://draw.io>