

Entwicklung eines Server- Prototypen für Saros

Verteidigung meiner Bachelorarbeit

Übersicht

- Einführung
- Planung
- Implementierung
- Demo
- Ausblick

Einführung

Worum geht's?

- Eine Software (Saros) soll ein neues Feature (eine Serverfunktion) bekommen
- Dieses ist zu groß, um an einem Stück entwickelt zu werden
- Stattdessen: Entwicklung eines Prototypen mit den wichtigsten Features
- Zudem: Sammeln von Erfahrungen, die bei der Weiterentwicklung helfen können

Was ist Saros?

- Plug-In für Eclipse
- Verteilte, kollaborative Softwareentwicklung
- Arbeiten in Sitzungen
 - Geleitet von einem „Host“

Problematik/Motivation

- Zeitversetztes Arbeiten erschwert
- Host kann Sitzung nicht verlassen

Planung

Vision

- Server verwaltet laufende Sitzungen
- Benutzer können sich einklinken
- Beim Verlassen der Sitzung bleibt diese erhalten

Vorüberlegungen

- Stand-Alone oder Teil des Plug-Ins
- Anzahl der Sitzungen pro Server
- „Proof of Concept“ oder Alpha-Version
- Kontinuierliche Integration

Ausgewählte Features

- Aktivieren/Deaktivieren des Servers
- Bekanntmachen des Servers
- Sitzungsstart
- Sitzungsbeitritt
- Einladen von Personen
- Abfrage von Sitzungsinformationen

Implementierung

Feature-Toggle

- Boolean, über den Features an- und ausgeschaltet werden können
- Von mir verwendet, um Entwicklung nicht zu stören

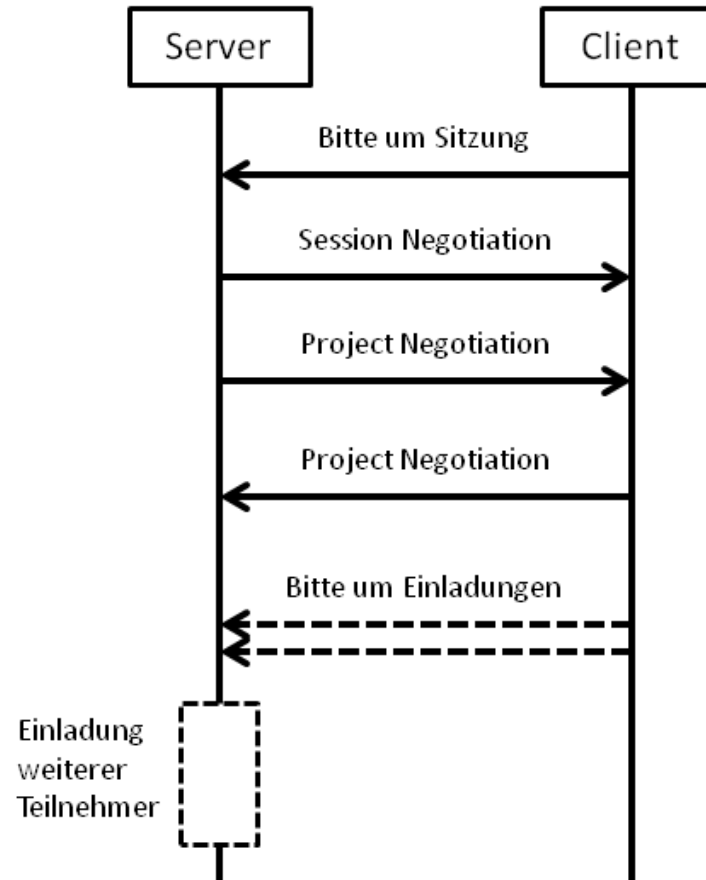
Aktivierung des Server-Features

- Erreichbar über Saros-Einstellungen im Eclipse-Menü
- Speicherung in „PreferenceStore“ von Eclipse

Bekanntmachen des Servers

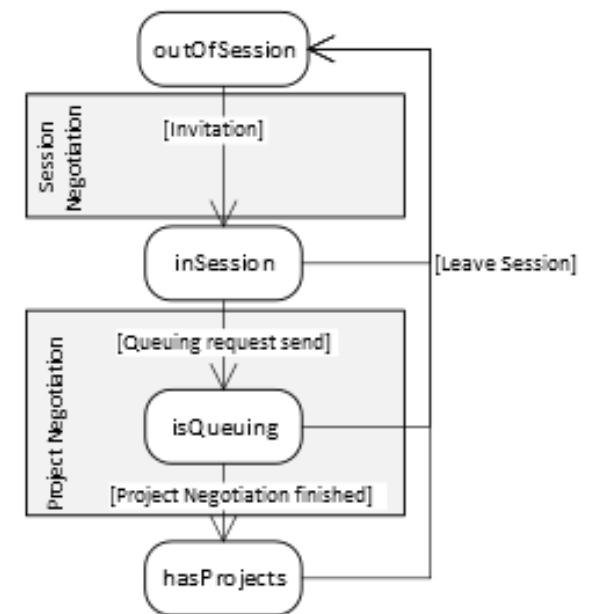
- Server an XMPP-Erweiterung erkennbar
- Clients können über „Service Discovery“ die Server ihren Kontakten finden

Sitzungsstart (ursprünglicher Plan)

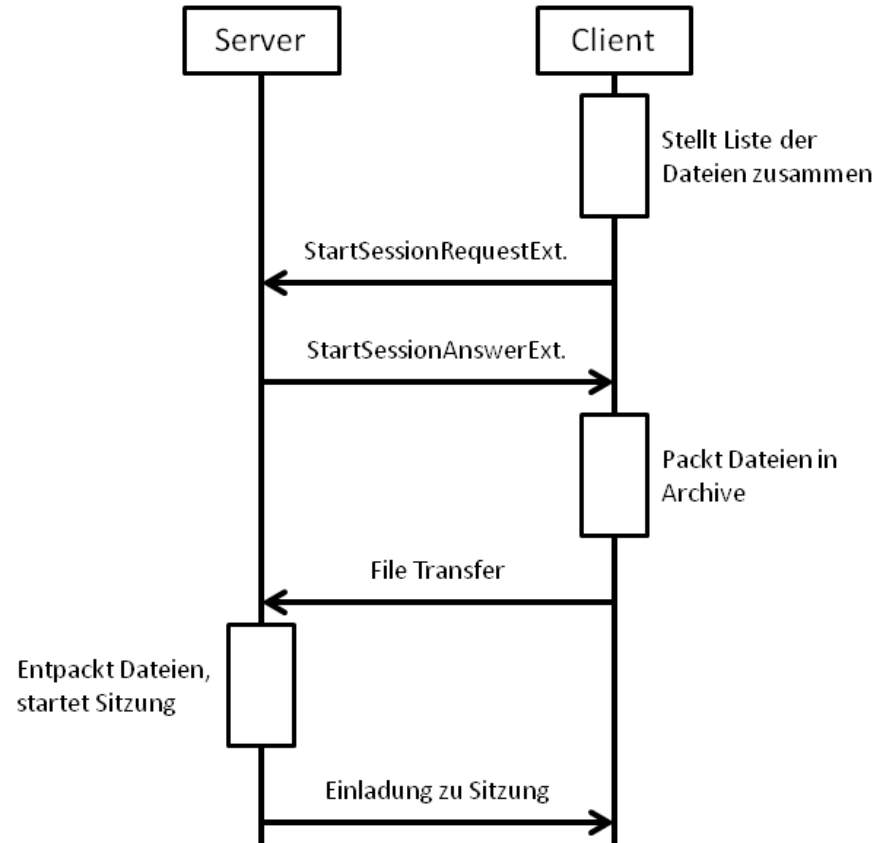


Sitzungsstart (ursprünglicher Plan)

- Einführung einer Nachricht, in der der Client den Server um den Start einer Sitzung und eine Einladung dazu bittet
- Kritik von Entwicklern:
 - Verletzung der User States
 - Verletzung der Trennung zwischen Server und Client



Server Transfer Negotiation



Sitzungsbeitritt

- Ursprünglich geplant, dieselbe Nachricht wie zum Sitzungsstart zu verwenden
- Nach der Einführung der Server Transfer Negotiation ist diese Doppelverwendung nicht mehr nötig
- Mechanismus:
 - Benutzer bittet um Einladung
 - Server prüft, ob Sitzung läuft
 - Server lädt ein

Einladung zu einer Sitzung

- Client bittet Server darum, einen Nutzer einzuladen
- Server lädt ein

Abfrage von Sitzungsinformationen

- Intention: Nutzer muss wissen, ob er in eine Sitzung eintreten kann. Hilfreich ist, den Inhalt der Sitzung zu kennen.
- Geplant: Server teilt auf Anfrage mit:
 - Ob eine Sitzung läuft
 - Namen der geteilten Projekte
 - Namen der Teilnehmer
- Nach Kritik bzgl. des Datenschutzes habe ich die Namen der Teilnehmer entfernt

Nicht geschafft

- Server Transfer Negotiation unvollendet
- Wizard zum Sitzungsstart unvollendet
- Wizard zum Sitzungsbeitritt nicht umgesetzt
- Einladungen zu Sitzungen nicht ins Hauptprojekt integriert

Demo

Ausblick

Mögliche Erweiterungen (Auswahl)

- Projekte aus laufender Sitzung entfernen
- Mehrere Sitzungen pro Server erlauben
- Stand-Alone-Server