

Vortrag III: Abschluss Entwicklung einer GUI für den gMix-Simulator

Malte Weinschenk, Jörg Langnickel,
Jan Carsten Lohmüller & Alexander Beifuß

24. Januar 2014

Inhalt

- 1 Einleitung
- 2 GUI
- 3 Architektur
- 4 Annotations
- 5 Dependency Checker
- 6 Live Demo
- 7 Letzter Entwicklungsstand
- 8 Aktueller Entwicklungsstand (Live Demo)
- 9 Ausblick auf geplante Features
- 10 Diskussion

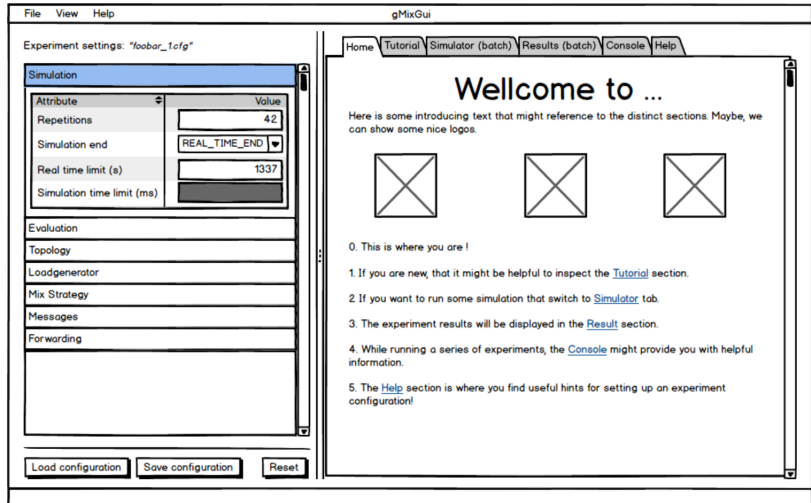
Ausgangssituation

Motivation

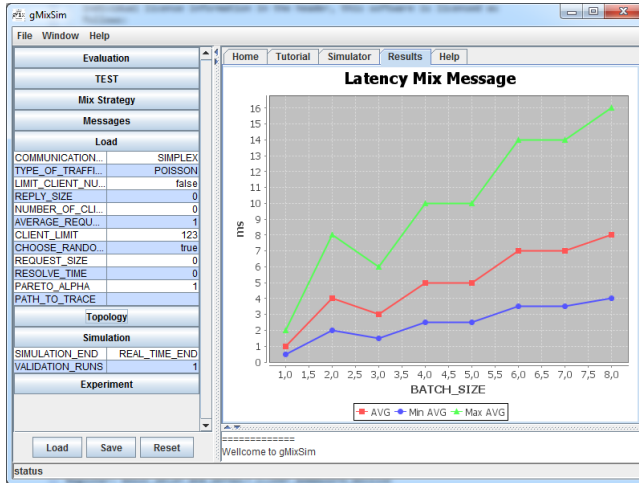
Benutzergruppen

Ziele

Designidee



Erste Umsetzung



gMixGUI

File Edit Window Help

Plugin Configuration

- ☒ Underlay-net
- ☐ Topology
- ☒ Mix Server
- ☐ DISTINCT_USER_BATCH
- Maximum reply delay**
100
- ☒ Mix Proxy
- ☒ Recoding Scheme
- ☒ Mix Client
- ☒ Load Generator
- Properties To Vary**

- General Configuration**
 - ☒ Simulation
 - ☒ Plottype

Load Save Reset

Configuration Selection

- etc/experiments/example_old.cfg
- etc/experiments/experiment.cfg
- etc/experiments/new.cfg

* Multiple selection is possible

Simulation Control

Start Simulation
Stop Simulation
Clear Results

Export Results

Export Graph

Simulation Status

Console

```

2014-01-23 21:16:00 Item in property not found: CONSOLEFRAME_XPOS
2014-01-23 21:16:00 Using default value for item CONSOLEFRAME_XPOS
2014-01-23 21:16:00 Item in property not found: CONSOLEFRAME_YPOS
2014-01-23 21:16:00 Using default value for item CONSOLEFRAME_YPOS
2014-01-23 21:16:00 Item in property not found: CONSOLEFRAME_WIDTH
2014-01-23 21:16:00 Using default value for item CONSOLEFRAME_WIDTH
2014-01-23 21:16:00 Item in property not found: CONSOLEFRAME_HEIGHT
2014-01-23 21:16:00 Using default value for item CONSOLEFRAME_HEIGHT
2014-01-23 21:16:01 Item in property not found: GUISERVICE_TOGGLE_HOME_TAB
2014-01-23 21:16:01 Using default value for item GUISERVICE_TOGGLE_HOME_TAB
2014-01-23 21:16:01 Can not read value for LSB_REPLY_RATE
2014-01-23 21:17:11 Finished simulator with results
    
```

Results

Experiment 0

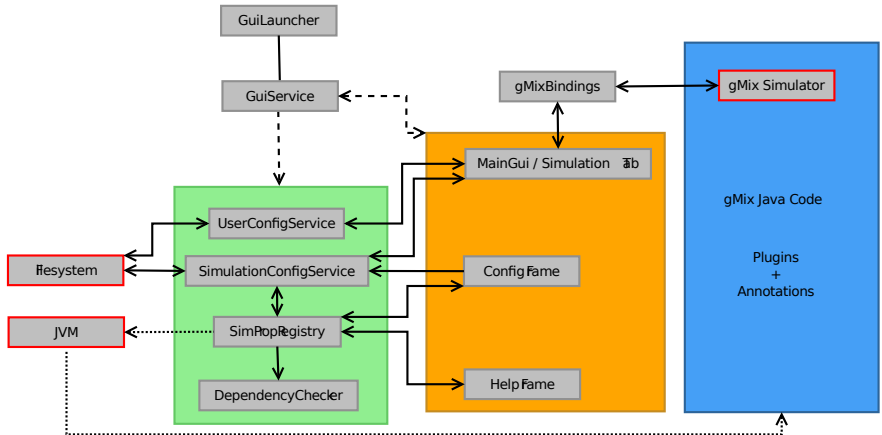
plot_clientLatencyMixMessage - effect of BATCH_SIZE

ms

BATCH_SIZE

MED IN AVG (AVG_CLIENT_LATENCY_REQUESTMESSAGE)
 MIN AVG (AVG_CLIENT_LATENCY_REQUESTMESSAGE)
 MAX AVG (AVG_CLIENT_LATENCY_REQUESTMESSAGE)
 MED MAX (MAX_CLIENT_LATENCY_REQUESTMESSAGE)
 MIN MAX (MAX_CLIENT_LATENCY_REQUESTMESSAGE)
 MAX MAX (MAX_CLIENT_LATENCY_REQUESTMESSAGE)

Architektur



Motivation für Annotations

	Annotations	XML zentr.	XML dezent.	Statisch
Plugin Struktur	++	-	++	--
Initialer Aufwand	-	+	+	++
Aufwand neues Plugin	++	+	+	--
Erweiterbarkeit Fkt.	+	+	+	?
Unterstützung d. IDE	++	+	+	++

Für GUI-Benutzer ist die verwendete Technik transparent.

Plugin-Entwickler profitieren jedoch sehr von den Annotations.

⇒ Gedanken zur GUI nur in Sonderfällen nötig!

⇒ Rapid Prototyping

Der Mehraufwand bei der Programmierung des Frameworks ist gerechtfertigt.

Dependency Checker

Live Demo

Nutzergruppen

1 Nutzer in der Lehre

- Das System muss einfach zu bedienen sein
- Vermeidung einer überladenen GUI mit zu viel Details
- Unterstützung durch die GUI bei der Fehlervermeidung
- Übersichtliche Präsentation der Ergebnisse

Nutzergruppen

1 Nutzer in der Lehre

- Das System muss einfach zu bedienen sein
- Vermeidung einer überladenen GUI mit zu viel Details
- Unterstützung durch die GUI bei der Fehlervermeidung
- Übersichtliche Präsentation der Ergebnisse

2 Nutzer in der Forschung

- Kontrolle über viele / alle Parameter
- Flexible Darstellung der Ergebnisse
- Stapelverarbeitung von Experimenten

Nutzergruppen

1 Nutzer in der Lehre

- Das System muss einfach zu bedienen sein
- Vermeidung einer überladenen GUI mit zu viel Details
- Unterstützung durch die GUI bei der Fehlervermeidung
- Übersichtliche Präsentation der Ergebnisse

2 Nutzer in der Forschung

- Kontrolle über viele / alle Parameter
- Flexible Darstellung der Ergebnisse
- Stapelverarbeitung von Experimenten

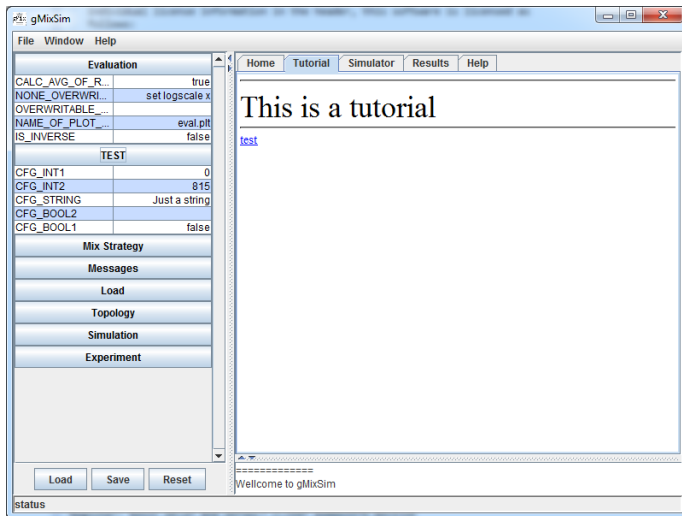
3 Plug-In Entwickler

- Möglichst wenig Aufwand bei der GUI
- Plug-in soll entkoppelt betrachtet werden
- Es sollen keine Namenskollisionen mit anderen Plug-ins entstehen

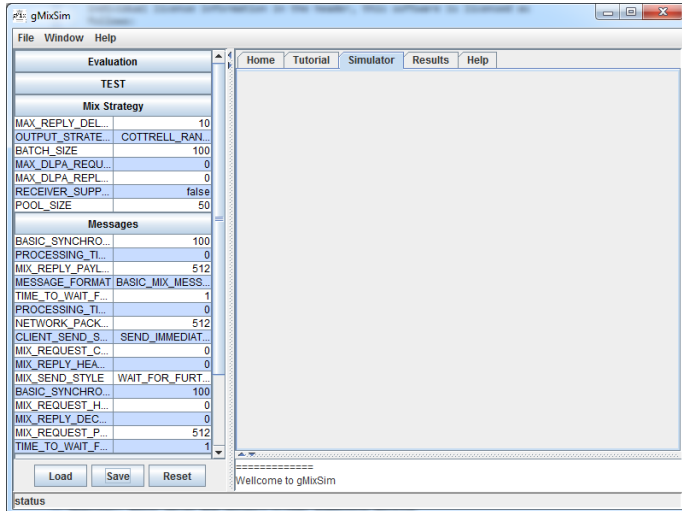
Willkommens Reiter



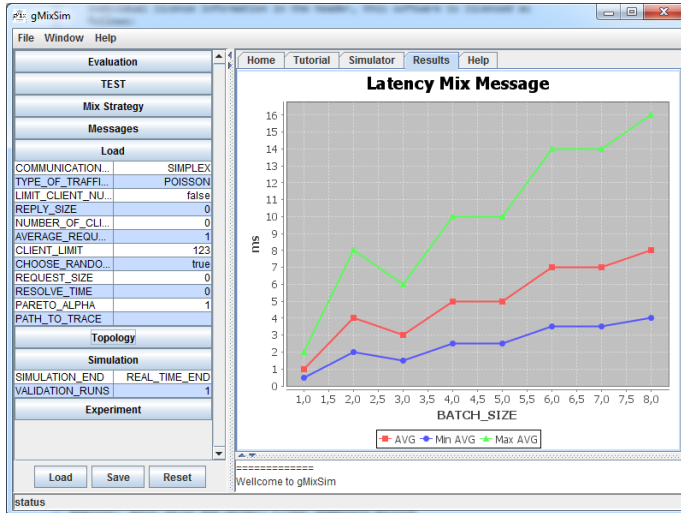
Tutorial Reiter



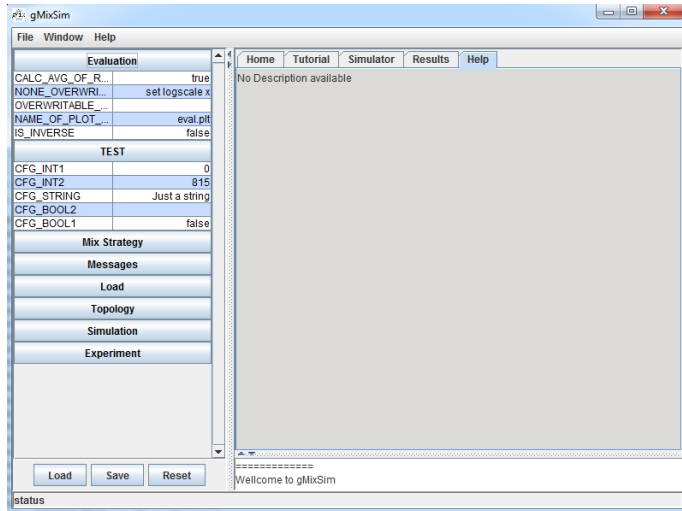
Simulator Reiter



Ergebnis Reiter



Hilfe Reiter



Aufgaben

Hauptkritikpunkte aus der letzten Besprechung

- 1 Fenster (Hilfe und Konfigurationswerkzeug) entkoppelbar machen
- 2 Hilfe und Tutorial in eins zusammenfassen
- 3 Simulator- und Ergebnis-Tabs zusammenfassen
- 4 Visuelle Unterstützung der Accordion-Elemente durch Icons

Fortschritt

Implementiert

- 1 Hilfe- und Konfigurationsfenster entkoppelbar
- 2 Hilfe und Tutorial zusammengefasst (basieren auf HTML)
- 3 Benutzer Konfigurationen bzgl. GUI können gespeichert werden
- 4 Abhängigkeiten zwischen Konfigurations-Attributen können festgelegt werden
- 5 Problematische Werte werden durch Fehlermeldung aufgezeigt
- 6 Plug-Ins können anhand des Classpaths eingelesen werden
- 7 Typspezifische Annotations für String, Bool, Int und Float
- 8 Simulator ist aus der GUI aufrufbar
- 9 Visuelle Unterstützung der Accordion-Elemente durch Icons

Live Demonstration

Ausblick auf geplante Features I

Quick Wins

- 1 Ausblenden des Home-Tabs ermöglichen
- 2 Laden und Speichern von Konfigurationen
 - Sowie Übergabe der Konfigurationen an den Simulator
- 3 Rendern der Simulationsergebnisse
 - In gesonderten Tabs oder entkoppelten Fenstern
- 4 Annotations auf das gMix-Projekt anwenden
- 5 GUI für den Simulator-Tab

Ausblick auf geplante Features II

Weitere Entwicklung der Annotations

1 Dependency-Violations

- Visuelle Unterstützung des Benutzers durch Anzeige von Abhängigkeiten (Feature Models)

2 Plugin-Annotations

- Modellierung von Plugin-Abhängigkeiten (Level 1 bis Level 5)

3 (Dependency-)Service-Plugin

- Ebenfalls Modellierung von Plugin-Abhängigkeiten

4 Dynamisches Erstellen der Hilfe basierend auf Annotations

- z.B. Minimum- und Maximumwerte aus den Annotations extrahieren

5 Grafische Elemente anstatt Attribut-Tabelle

- z.B. Spinner für Integer, Textbox für freie Strings, Dropdownmenu für vordefinierte Werte

Diskussion