

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

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

27. Januar 2014

Inhalt

- 1 Einleitung
- 2 GUI
- 3 Architektur
- 4 Annotations
- 5 Dependency Checker
- 6 Live Demo

Ausgangssituation gMix

- Last Generierung
 - Bestimmung von Lastcharakteristika
 - Modellierung der Clients
- Simulation & Messung
 - Modellierung der Mixe
 - Konfiguration der Plug-Ins
- Evaluation
 - Auswertungs Plug-Ins

Motivation

Benutzergruppen

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

Benutzergruppen

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

Benutzergruppen

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

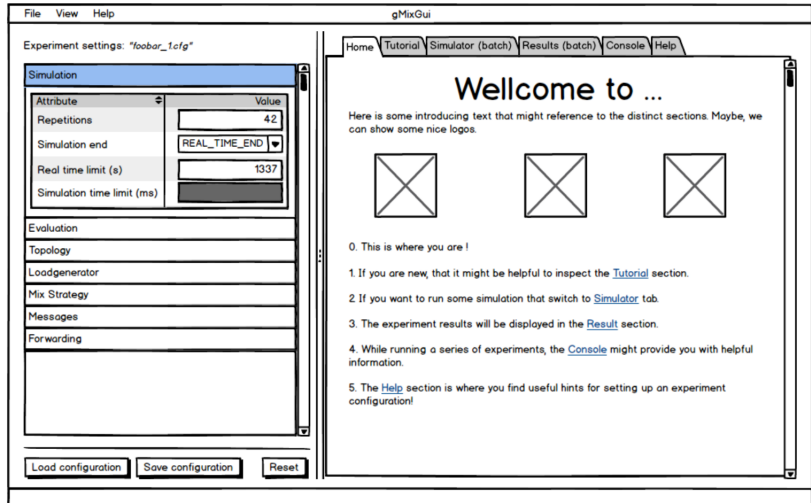
- Möglichst wenig Aufwand bei der GUI
- Plug-in soll entkoppelt betrachtet werden
- Es sollen keine Namenskollisionen mit anderen Plug-ins entstehen

Ziele

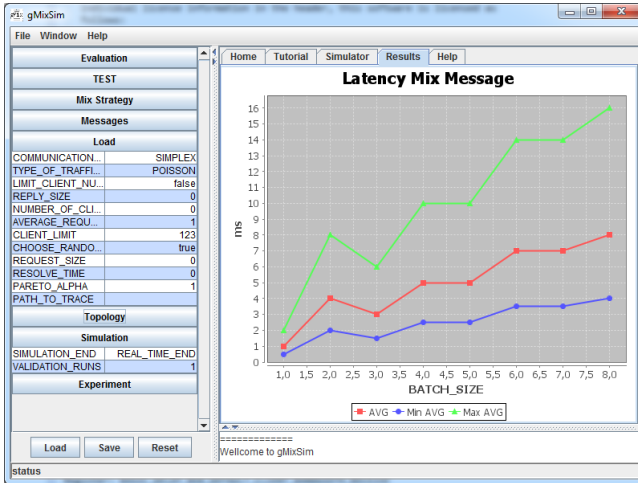
Projektziel:

- 1 Grafisches Werkzeug zur Erstellung von gMix-Konfigurationen
- 2 Augenmerk auf Anwenderfreundlichkeit
- 3 Augenmerk auf Entwicklerfreundlichkeit

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

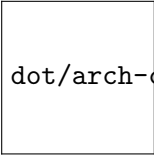
plot_clientLatencyMixMessage - effect of BATCH_SIZE

ms

BATCH_SIZE

MED IN AVG (AVG_CLIENT_LATENCY_REQUEST MIXMESSAGE) — +
 MIN AVG (AVG_CLIENT_LATENCY_REQUEST MIXMESSAGE) — +
 MAX AVG (AVG_CLIENT_LATENCY_REQUEST MIXMESSAGE) — +
 MIN MAX (MAX_CLIENT_LATENCY_REQUEST MIXMESSAGE) — x
 MAX MAX (MAX_CLIENT_LATENCY_REQUEST MIXMESSAGE) — x
 MED IN AVG (AVG_CLIENT_LATENCY_REQUEST MIXMESSAGE) — +
 MIN AVG (AVG_CLIENT_LATENCY_REQUEST MIXMESSAGE) — +
 MAX AVG (AVG_CLIENT_LATENCY_REQUEST MIXMESSAGE) — +
 MIN MAX (MAX_CLIENT_LATENCY_REQUEST MIXMESSAGE) — x
 MAX MAX (MAX_CLIENT_LATENCY_REQUEST MIXMESSAGE) — x

Architektur



dot/arch-crop.pdf

Motivation für Annotations

	Annotations	XML zentr.	XML dezentr.	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

- Konfigurationsdateien
- Keine Unterstützung von Abhängigkeiten und Wertebereichen
- Abhängigkeiten zwischen Properties:
 - Value Requirements z.B. Minimal- und Maximalwerte
 - Enable Requirements z.B. gegenseitiger Ausschluß
- Wertebereiche in einem Property: Minimal- und Maximalwerte
- Ziele:
 - Einfache Benutzung
 - Maximale Flexibilität
 - Wertebereiche in einem Property beachten

Live Demo