

GSM Simulation

GSMsim

Software Dokumentation

Version 1.0

Martin Figel
Matrikel-Nr.: 12561
Praktikum Softwaretechnik 1
Hochschule der Medien, Stuttgart

I Über dieses Dokument

Dies ist die Beschreibung der Software GSMsim, die im Zusammenhang mit einem Softwaretechnik 1 Praktikum von Martin Figel in Zusammenarbeit von DaimlerChrysler Services FleetBoard und der Hochschule der Medien entstanden ist.

II Entwicklungsstatus

Bearbeitungsstand: aktuell

Datum: 23.02.2006

Verfasser: Martin Figel, Hochschule der Medien

Projektname: GSM Simulation

III Änderungsübersicht

Nr.	Datum	Version	Geänderte Stellen	Grund	Bearbeiter
1	21.03.06	0.1	---	Erstversion	Figel
2	23.03.06	1.0	Freigabe		Figel

IV Inhalt

I	Über dieses Dokument	2
II	Entwicklungsstatus	3
III	Änderungsübersicht	4
IV	Inhalt	5
1	Anwendungszweck	6
2	Voraussetzungen und Annahmen	7
3	Funktionalität	8
3.1	Graphical User Interface	8
3.2	Messaging	8
3.3	Persistenz	8
3.4	Sonstige Funktionalität	8
4	Komponenten	9
5	Technologie	10
6	Installation (Windows 2000/XP)	11
6.1	Installieren der Java-Communication-API	11
6.2	Installation der restlichen Files	11
7	Bedienung	12
7.1	Starten des Programmes	12
7.2	Short-Message Panel	12
7.3	Transporting-Management	13
7.4	Vehicle Management	14
7.5	Message-Board	15
7.6	TestSzenario-Board	16
7.7	Communication-Panel	18
8	Ausblick	20
A	Abkürzungen	21
B	API-Spezifikation	22
C	Klassendiagramme/UML	23

1 Anwendungszweck

Weiterführende Automatisierung von Softwaretests, um Ressourcen (Personal, Zeit, Kosten für Kommunikation) einzusparen.

Dies wird hier durch die Realisierung eines Tools erreicht, welches mit der DaimlerChrysler-Telematik-Plattform kabelgebunden kommuniziert und GSM-Daten (SMS) zu Testzwecken simuliert.

Das GSM-Modem in der DCTP wird durch das Simulationstool, welches die notwendigen Informationen aus dem GSM-Netz bereitstellt, ersetzt.

Die Anwendung ist in der Lage, die Testeinheit (DispoPilot oder DCTP) mit GSM-Daten zu versorgen. Diese Daten können in verschiedenen Formaten vorliegen und werden durch die Anwendung erzeugt. Genauso können Nachrichten als verkettete SMS eingespielt bzw. empfangen werden. Das Simulationstool kann Nachrichten auf die korrekte Verarbeitung der Testeinheit hin verifizieren.

2 Voraussetzungen und Annahmen

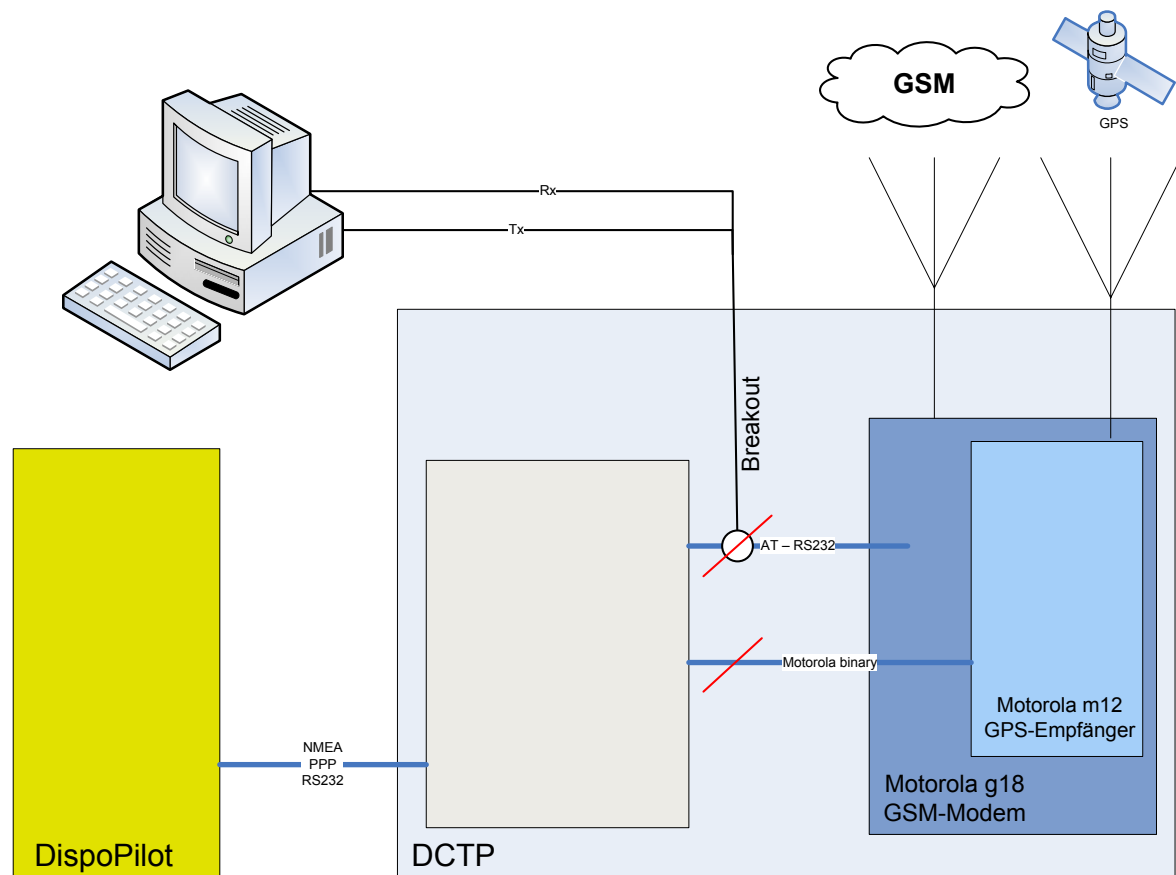
Folgende Hardware muss zur Verfügung stehen:

- 1 DaimlerChrysler Telematik Plattform (DCTP) mit Breakout-Platine

Optional:

- 1 DispoPilot
- 1 Cradle

Abbildung: Aufbau Breakout-Platine der DCTP



3 Funktionalität

3.1 Graphical User Interface

- Eine Grafische Benutzeroberfläche (GUI) erlaubt die Interaktion des Users mit der Anwendung
- Die Software ist für einen Testingenieur völlig transparent. Das heißt der Benutzer kann die Software nutzen ohne aufwendige Konfigurationen durchführen zu müssen.
- Es ist für einen Testingenieur eine intuitive GUI zur Bedienung der Anwendung vorhanden

3.2 Messaging

- Die Anwendung ist in der Lage Transporting- und Vehicle-Nachrichten zu generieren.
- Es ist möglich diese Nachrichten in einem konformen Short-Message-Format an die DCTP zu schicken.
- Es ist möglich verkettete Short-Messages zu verschicken.
- Für die Anwendung ist es möglich, eine von der DCTP erzeugte Short-Message zu empfangen.
- Einkommende Nachrichten können angezeigt werden.
- Abgespeicherte Nachrichten können einzeln verschickt werden.
- Aus den gespeicherten Nachrichten ist es möglich ein Testszenario zusammenzustellen, welches dann per Knopfdruck, die in dem Testszenario enthaltenen Nachrichten, einzeln mit Angabe einer Latenz an die DCTP verschickt.

3.3 Persistenz

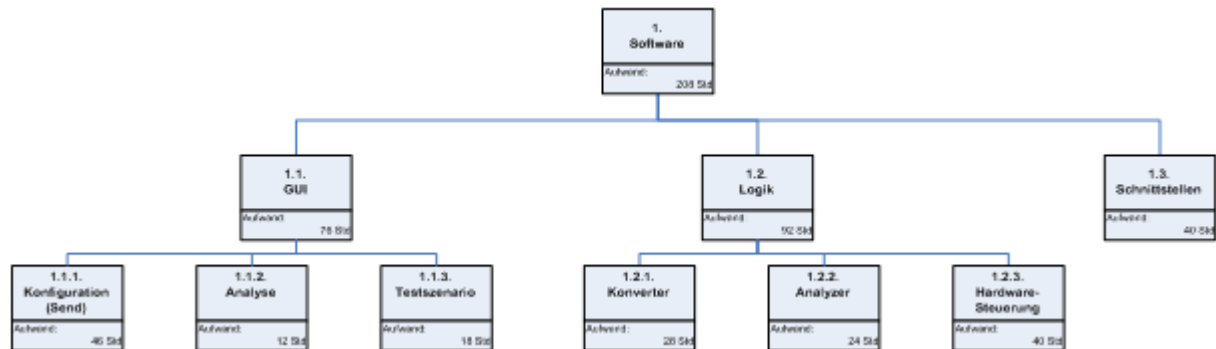
- Es ist möglich einzelne Nachrichten abzuspeichern.
- Es ist möglich einzelne Test-Szenarios abzuspeichern
- Short-Message-Konfigurationen können geladen werden.
- Es werden editierbare Config-Files zur Verfügung gestellt die dazu dienen eine möglichst hohe Flexibilität zu gewährleisten.

3.4 Sonstige Funktionalität

- Die Anwendung ist in der Lage der DCTP ein funktionierendes GSM-Modem „vorzugaukeln“.
- Über Schnittstellen ist es für spätere Integration möglich, einfachst eine neue GUI auf der vorhandenen Logik aufzusetzen.
- Die Anwendung ist in der Lage die Kommunikation über RS-232 abzuwickeln.
- Die Anwendung kommuniziert performancetechnisch auf dem Niveau des normalerweise verbauten Modems
- Die Anwendung erlaubt Langzeittests

4 Komponenten

Die einzelnen Komponenten der Software



5 Technologie

- Strenge Aufsplittung der Anwendung nach dem Model – View – Controller Design Pattern
 - Flexibles Programmdesign
 - Individuelle GUI kann ohne weiteres erstellt werden
 - Übersichtlichkeit ist gegeben für spätere Erweiterung
- Focus auf die technische Umsetzung, weniger auf Design
 - Bedienung durch erfahrene Testingenieure ;)
- Realisierung der Kommunikation über die Java Communications API

6 Installation (Windows 2000/XP)

6.1 Installieren der Java-Communication-API

Da standardmäßig im JDK kein Zugriff auf die serielle Schnittstelle als Bibliothek angeboten wird, installieren Sie bitte vorab nach folgendem Muster die Java-Communication-API:

Kopiere die `win32com.dll` in das `<jdk>\jre\bin` Verzeichnis.

Kopiere die `comm.jar` nach `<jdk>\jre\lib\ext`.

Kopiere die `javax.comm.properties` nach `<jdk>\jre\lib`.

- ➔ „<jdk>“ ist Platzhalter für ihr Installationsverzeichnis ihres Java-SDK.
- ➔ **Achtung! Sollten sie ein Java-Update durchführen und danach ein anderes JRE im Klassenpfad stehen, gehen sie wie oben erneut vor!**
- ➔ **Bitte ändern sie nicht den Klassenpfad!**

6.2 Installation der restlichen Files

Kopieren folgenden Files in ein Verzeichnis Ihrer Wahl:

<code>GSMsim.jar</code>	-> die ausführbare Datei
<code>default.ini</code>	-> enthält die default-Werte für die Initialisierung
<code>Messages.csv</code>	-> abgespeicherte Standard-Messages
<code>TestSzenario.csv</code>	-> ein abgespeichertes Test-Szenario

Für die Dokumentation der API kopieren sie sich das Verzeichnis „doc“ an einen Platz ihrer Wahl

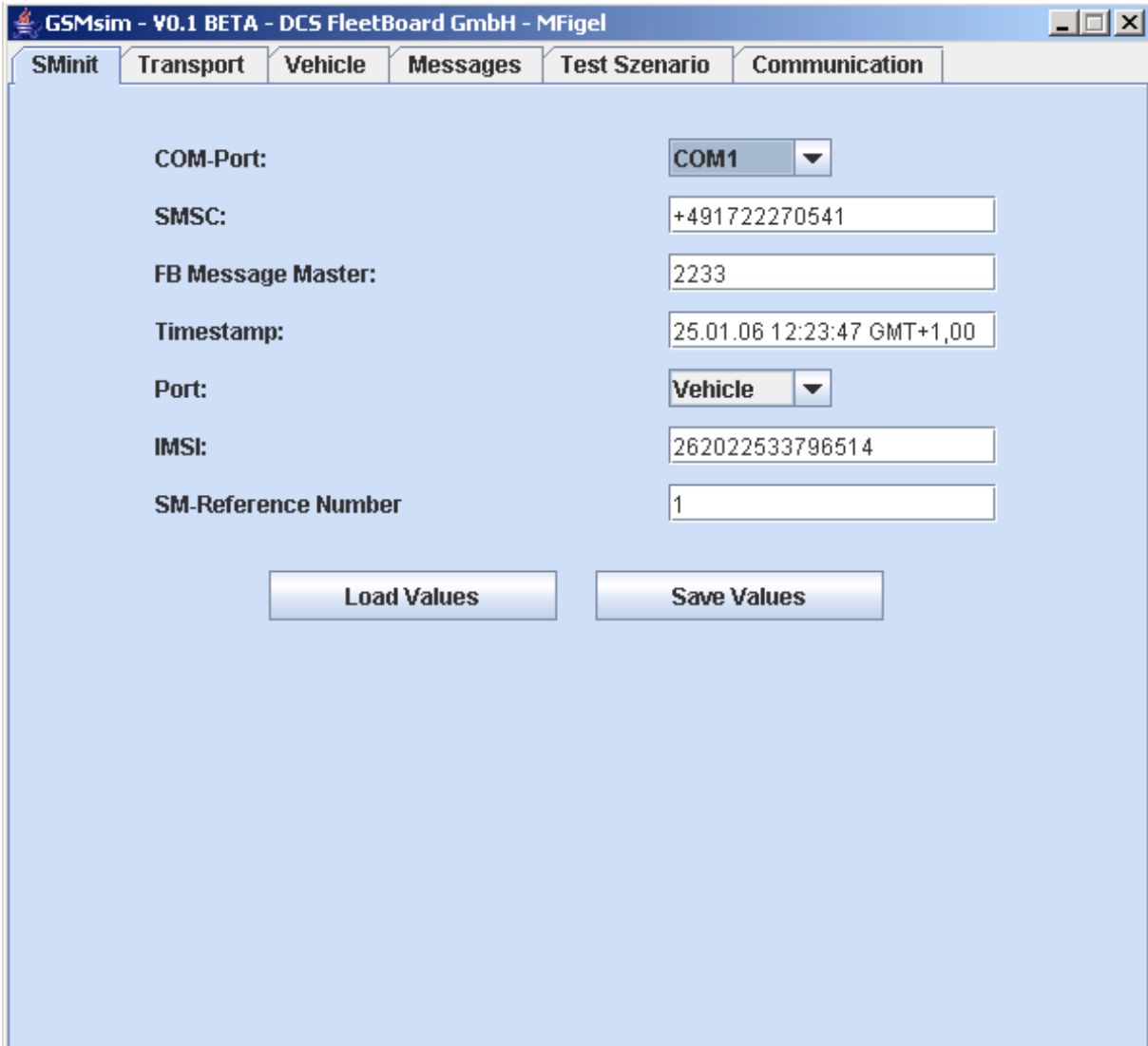
7 Bedienung

7.1 Starten des Programmes

Indem sie auf „GSMsim.jar“ doppelklicken starten sie die Anwendung.

7.2 Short-Message Panel

Nach dem starten der Anwendung befinden sie sich direkt im Konfigurations-Panel für die Short-Messages.



GSMsim - V0.1 BETA - DCS FleetBoard GmbH - MFigel

SMinit | Transport | **Vehicle** | Messages | Test Szenario | Communication

COM-Port:

SMSC:

FB Message Master:

Timestamp:

Port:

IMSI:

SM-Reference Number:

7.2.1 Einstellungen laden

Mit „Laod Values“ besteht die Möglichkeit Werte von der config.ini zu laden.

7.2.2 Einstellungen dauerhaft abspeichern

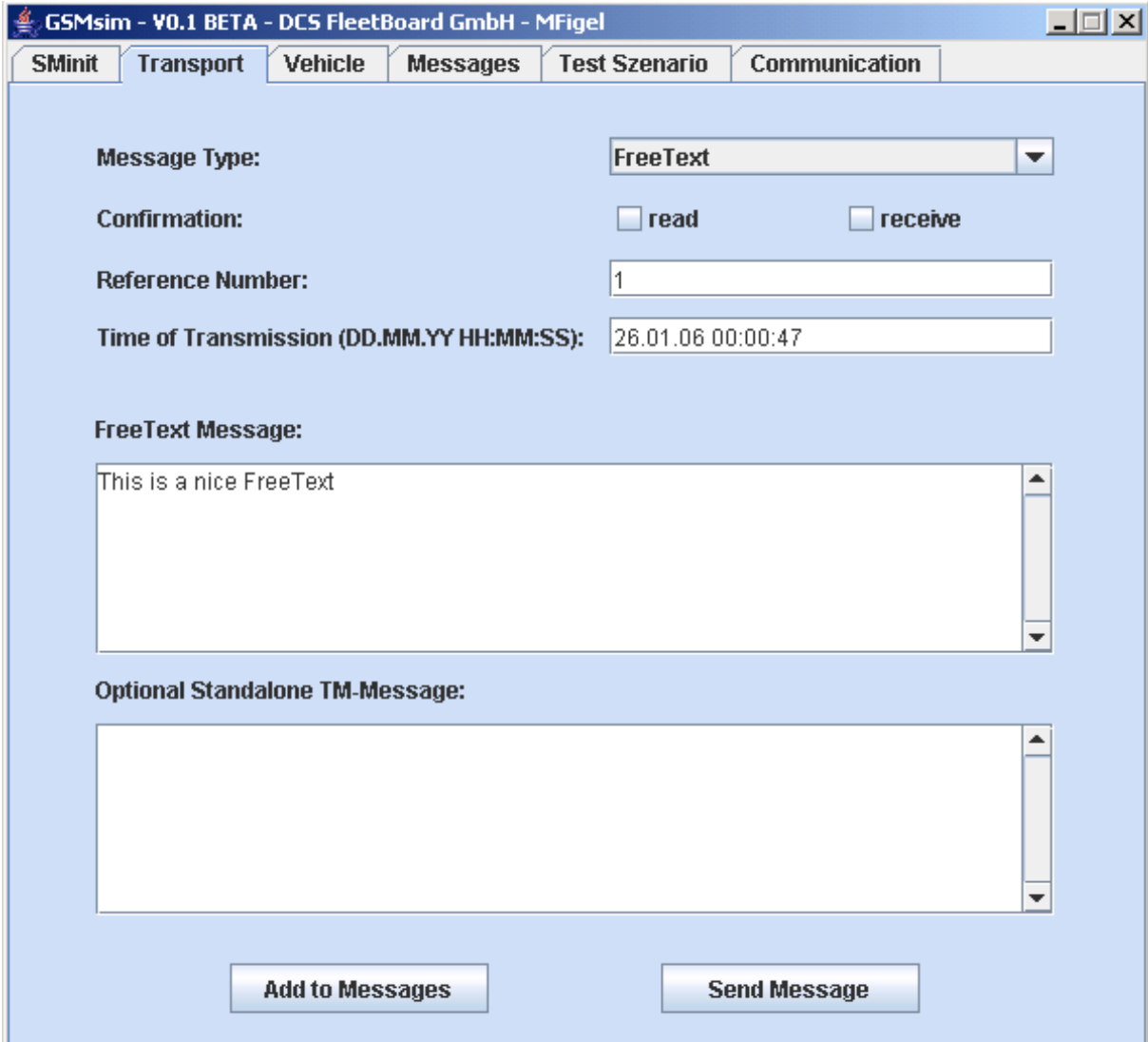
Über „Save Values“ (noch nicht implementiert) besteht die Möglichkeit die aktuell eingegebenen Werte dauerhaft abzuspeichern.

7.2.3 Serial-Port auswählen

Über die ComboBox „COM-Port“ lässt sich der zu verwendende Serial-Port einstellen.

7.3 Transporting-Management

Über den Reiter „Transport“ gelangen sie zum Panel um Nachrichten an den DispoPilot zu schicken.



The screenshot shows the 'Transport' tab of the GSMsim application. The interface includes the following elements:

- Message Type:** A dropdown menu set to 'FreeText'.
- Confirmation:** Two checkboxes labeled 'read' and 'receive', both of which are currently unchecked.
- Reference Number:** A text input field containing the number '1'.
- Time of Transmission (DD.MM.YY HH:MM:SS):** A text input field containing the timestamp '26.01.06 00:00:47'.
- FreeText Message:** A large text area containing the text 'This is a nice FreeText'.
- Optional Standalone TM-Message:** An empty text area for additional message content.
- Buttons:** Two buttons at the bottom: 'Add to Messages' and 'Send Message'.

7.3.1 Nachrichten zum Message-Board hinzufügen

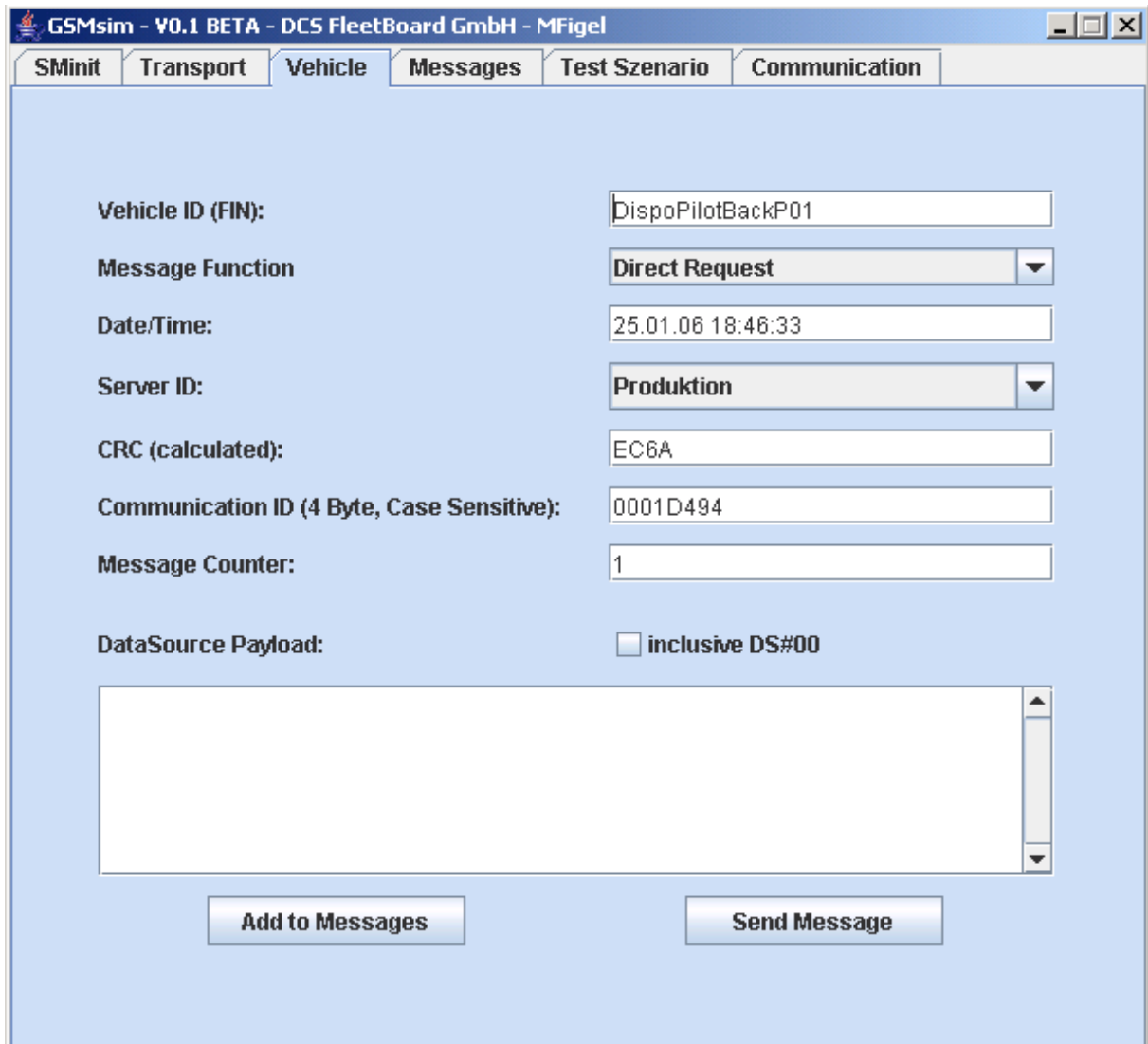
Über den Button „Add to Messages“ lässt sich eine hier definierte Nachricht zum Message-Board hinzufügen. -> siehe auch 7.5 Message-Board

7.3.2 Nachricht senden

Über den Button „Send Message“ läßt sich eine hier definierte Nachricht direkt über die DCTP an den DispoPilot schicken.

7.4 Vehicle Management

Über den Reiter „Vehicle“ gelangen sie zum Panel um Nachrichten an die DCTP zu schicken.



GSMsim - V0.1 BETA - DCS FleetBoard GmbH - MFigel

SMinit | Transport | **Vehicle** | Messages | Test Szenario | Communication

Vehicle ID (FIN):

Message Function:

Date/Time:

Server ID:

CRC (calculated):

Communication ID (4 Byte, Case Sensitive):

Message Counter:

DataSource Payload: inclusive DS#00

7.4.1 Nachrichten zum Message-Board hinzufügen

Über den Button „Add to Messages“ läßt sich eine hier definierte Nachricht zum Message-Board hinzufügen. -> siehe auch 7.5 Message-Board

7.4.2 Nachricht senden

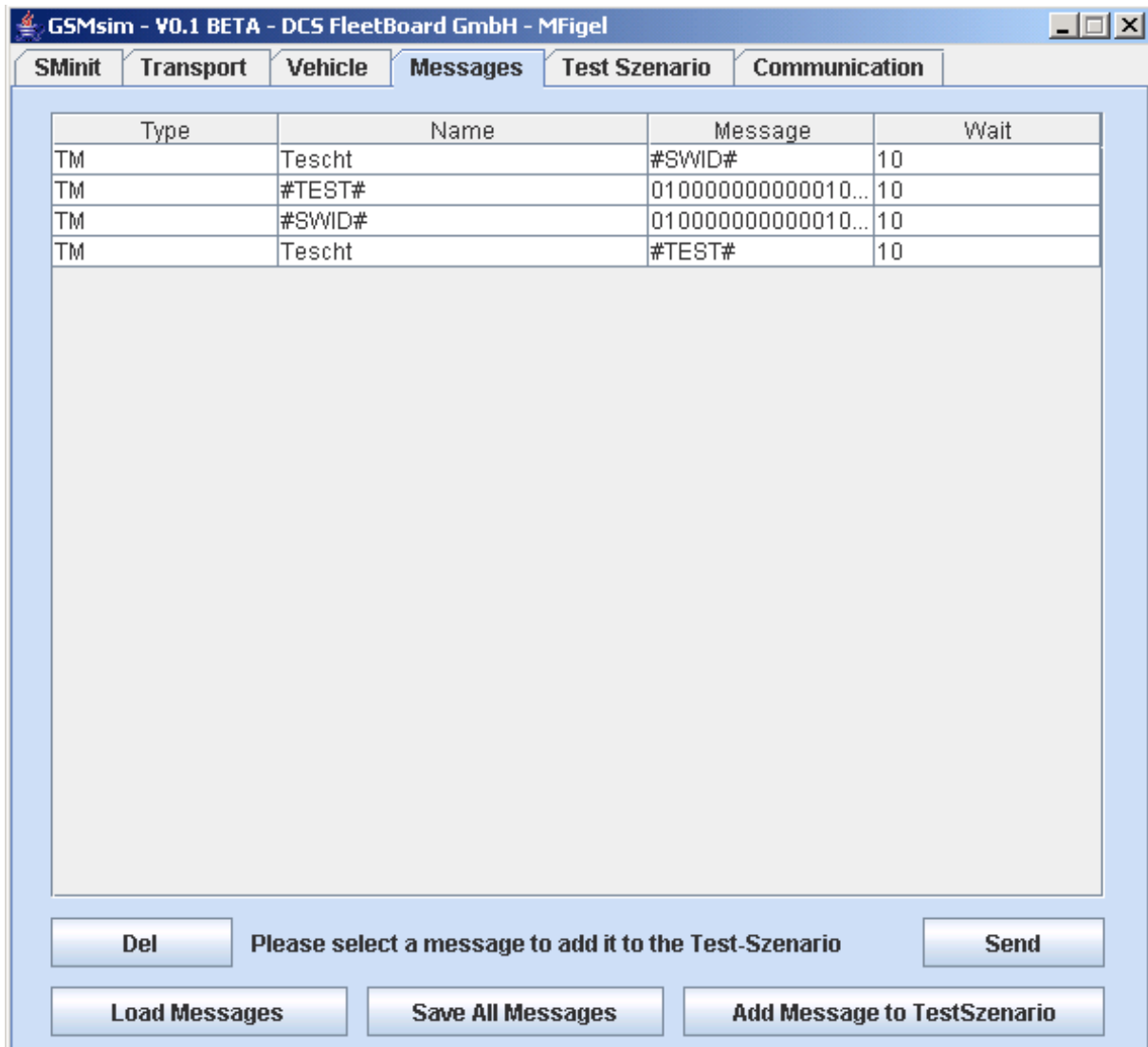
Über den Button „Send Message“ läßt sich eine hier definierte Nachricht direkt an die DCTP schicken.

7.5 Message-Board

Über den Reiter „Messages“ gelangen sie zum Message-Board indem gespeicherte Nachrichten verwaltet werden können.

7.5.1 Die einzelnen Spalten der Tabelle:

- Type -> gibt an ob es sich um eine SM oder TM-Nachricht handelt.
- Name -> vergeben sie hier einen aussagekräftigen Namen für ihre Nachricht
- Message -> Die Payload der Nachricht
- Wait -> Zeit (in sec) bis zum senden der nächsten Nachricht



Type	Name	Message	Wait
TM	Tescht	#SWID#	10
TM	#TEST#	0100000000000010...	10
TM	#SWID#	0100000000000010...	10
TM	Tescht	#TEST#	10

Buttons: Del, Please select a message to add it to the Test-Szenario, Send, Load Messages, Save All Messages, Add Message to TestSzenario

7.5.2 Nachrichten löschen

Markieren sie mit der Maus die Nachrichten die sie löschen möchten. Indem sie auf den „Del“-Button drücken werden die Nachrichten gelöscht.

7.5.3 Nachrichten laden

Über den Button „Load Messages“ können Nachrichten von der Datei „Messages.csv“ geladen werden.

7.5.4 Nachrichten speichern

Über den Button „Save All Messages“ können Nachrichten in der Datei „Messages.csv“ gespeichert werden.

7.5.5 Nachrichten senden

Markieren sie mit der Maus die Nachrichten die sie senden möchten. Über den Button „Send“ werden die Nachrichten nacheinander an die DCTP geschickt.

7.5.6 Nachrichten zu einem Testszenario hinzufügen

Über den Button „Add Message to TestSzenario“ lassen sich indem sie Zeilen markieren Nachrichten zu einem Testszenario hinzufügen.

7.5.7 Nachrichten ändern

Indem sie einfach auf das zu ändernde Feld klicken und einen neuen Wert eingeben können sie Werte verändern. Wichtig: Hinterher mit Enter- oder Return-Taste bestätigen.

7.5.8 #TEST# oder #SWID# hinzufügen

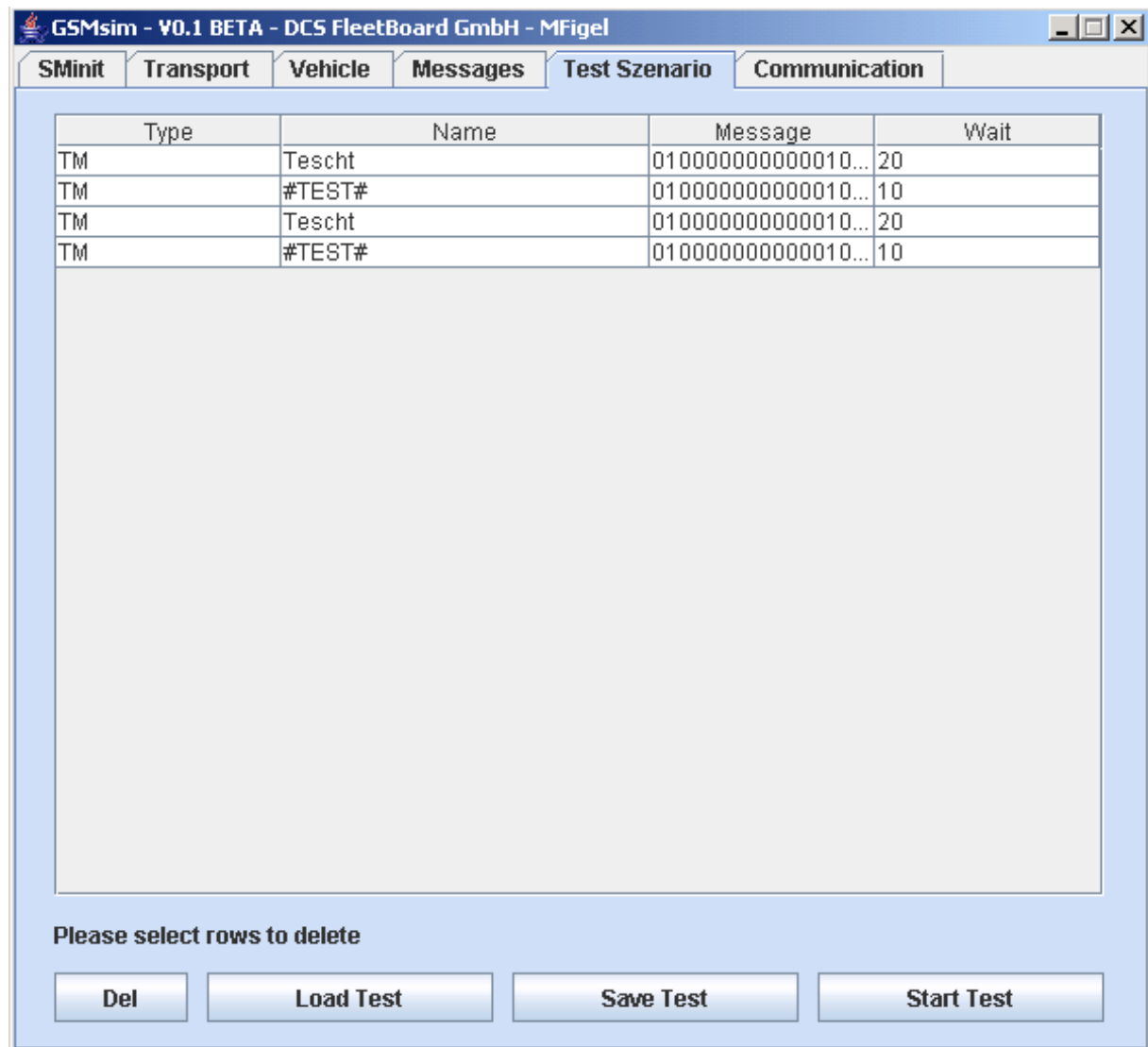
Editieren sie eine vorhande Nachricht indem sie bei „Name“ oder „Message“ entweder #TEST# oder #SWID# eingeben und mit der Enter- oder Return-Taste bestätigen.

7.6 TestSzenario-Board

Über den Reiter „Test Szenario“ gelangen sie zum TestSzenario-Board indem ein einzelner Test verwaltet werden kann.

7.6.1 Die einzelnen Spalten der Tabelle:

Type	->	gibt an ob es sich um eine SM oder TM-Nachricht handelt.
Name	->	vergeben sie hier einen aussagekräftigen Namen für ihre Nachricht
Message	->	Die Payload der Nachricht
Wait	->	Zeit (in sec) bis zum senden der nächsten Nachricht



7.6.2 Nachrichten löschen

Markieren sie mit der Maus die Nachrichten die sie löschen möchten. Indem sie auf den „Del“-Button drücken werden die Nachrichten gelöscht.

7.6.3 Test laden

Über den Button „Load Test“ kann ein Test von der Datei „TestSzenario.csv“ geladen werden.

7.6.4 Test speichern

Über den Button „Save Test“ kann ein Test in der Datei „Messages.csv“ gespeichert werden.

7.6.5 Nachrichten senden/Test starten

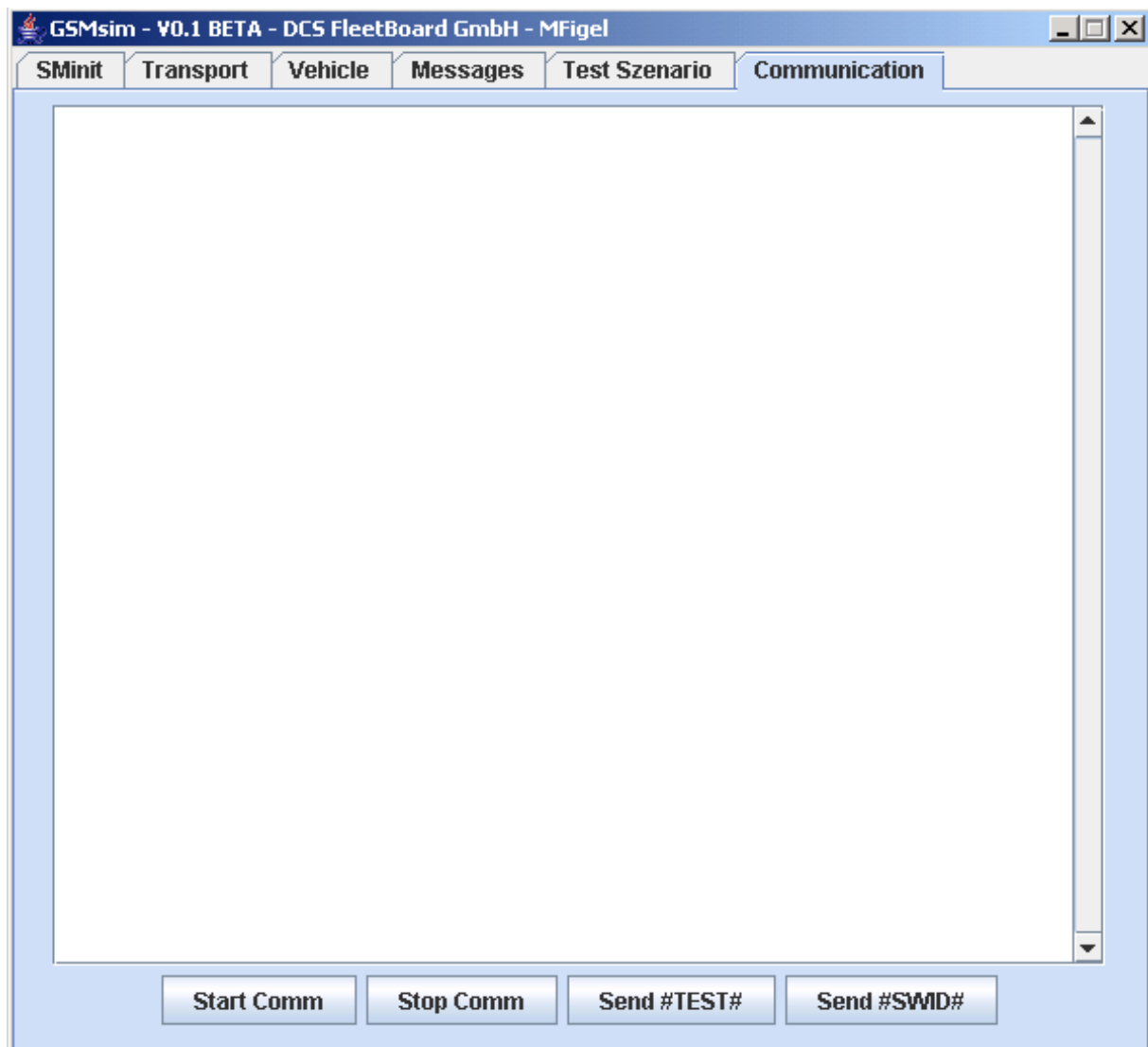
Über den Button „Start Test“ werden die Nachrichten nacheinander an die DCTP geschickt.

7.6.6 Nachrichten ändern

Indem sie einfach auf das zu ändernde Feld klicken und einen neuen Wert eingeben können sie Werte verändern. Wichtig: Hinterher mit Enter-/oder Return-Taste bestätigen.

7.7 Communication-Panel

Über den Reiter „Communication“ gelangen sie zum Communication-Panel indem sie ein- und ausgehende Nachrichten im mittigen Text-Feld angezeigt bekommen.



7.7.1 Kommunikation starten

Über den Button „Start Comm“ wird die DCTP-Kommunikation initialisiert und aufgebaut.

7.7.2 Kommunikation stoppen

Über den Button „Stop Comm“ wird die Kommunikation mit der DCTP abgebaut und der COM-Port geschlossen.

7.7.3 Test-Nachricht #TEST# senden

Über den Button „Send #TEST#“ wird die Nachricht #TEST# an die DCTP geschickt.

7.7.4 Test-Nachricht #SWID# senden

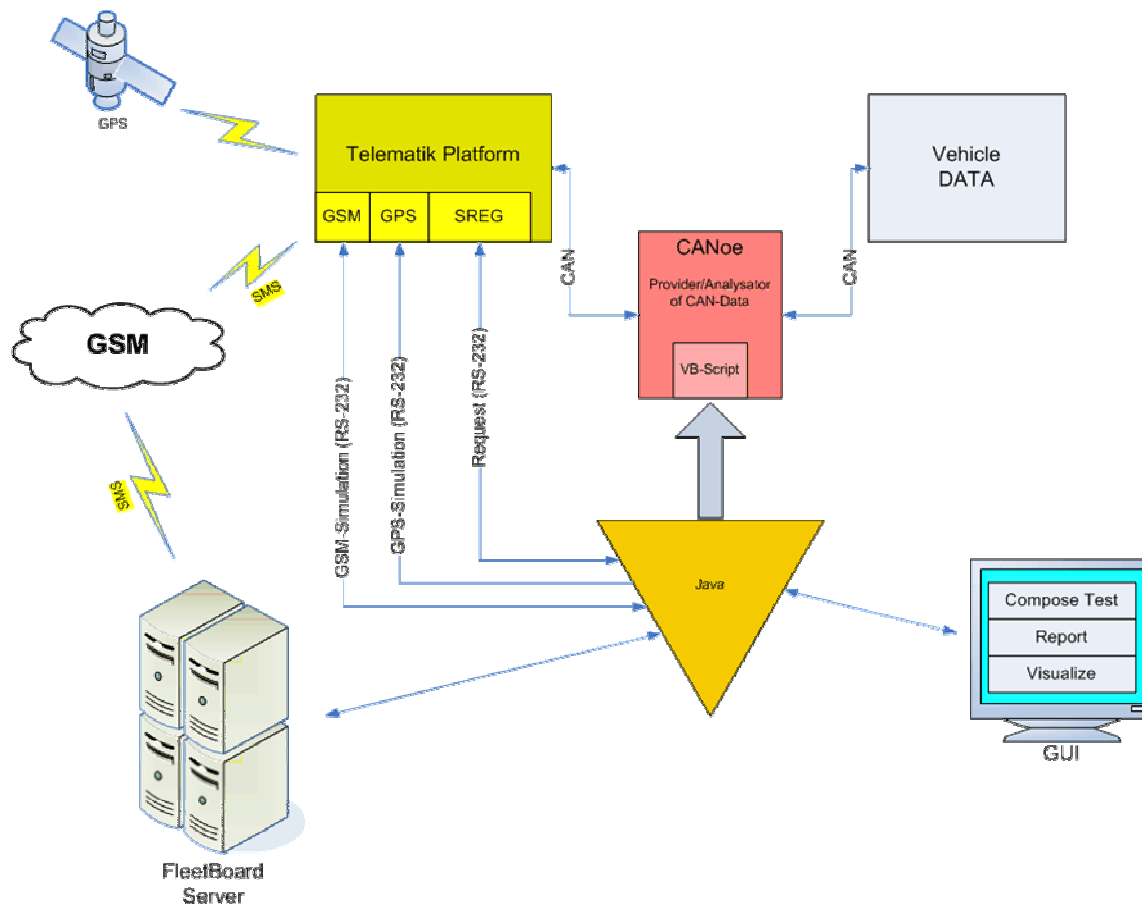
Über den Button „Send #SWID#“ wird die Nachricht #SWID# an die DCTP geschickt.

8 Ausblick

Da diese Software als Teil einer kompletten Testumgebung entwickelt wurde, besteht noch Bedarf für folgende Erweiterungen:

- SMS-Konfigurationen speichern
- Analyse der Antworten seitens der Testeinheit
- Simulation von CAN- und GPS-Messages
- Implementierung von grafischen Komponenten zur Visualisierung der Testabläufe
- Überführung der Testergebnis-Daten in ein universelles Format zur Weiterverarbeitung und Auswertung wie z.B. CSV, XML.
- Implementierung eines Reporting-Systems zur Auswertung der einzelnen Testfälle

Abbildung: Die GSM-Simulation als Teil einer umfassenden Testsuite



A Abkürzungen

Ausdruck/Abkürzung	Definition
TM	Transport-Management
VM	Vehicle-Management (Protokoll 3.1)
DCTP	DaimlerChrysler Telematik Plattform
SM	Short-Message
PRA	Proposal Risk Analysis
PID	Project Initiation Document

B API-Spezifikation

Siehe JavaDoc im „doc“-Verzeichnis

C Klassendiagramme/UML

Siehe JavaDoc im „doc“-Verzeichnis