

SMS-gestützte Datenbankanbindung

Patrick Fox

Mobile Computing Seminar, SS 2003

Zusammenfassung

Sms sind ein vergleichsweise günstiges und robustes Medium, um asynchron kurze Mitteilungen zu verschicken. Dadurch ist die Sms-Kommunikation prädestiniert, das Handy zur Fernbedienung des heimischen Rechners zu machen und mit einfachen Kommandos in einer Sms auf dem Heimatrechner Programme zu starten, die beispielsweise auf Internetanwendungen zugreifen oder angeschlossene Geräte steuern können.

Es wird zum einen erläutert, welche Hard- und Software man als Grundlage benötigt, um die Sms am heimischen Rechner zu empfangen. Zum anderen wird das in Java geschriebene Tool vorgestellt, welches die Verarbeitung der Sms übernimmt.

Als Anwendungsbeispiel wird "die Tippliga", ein internetbasiertes Bundesliga-Tippspiel für 18 Personen vorgestellt, das als Motivation diente, da den Benutzern ermöglicht werden sollte, ihre Tipps nicht nur über die Weboberfläche, sondern auch per Sms abgeben zu können.

Inhaltsverzeichnis

1	Motivation: die Tippliga	3
1.1	Modus der Tippliga	3
1.2	Logistik	3
2	Hardware	4
3	Software	4
3.1	Konvention des SMS-Formats	4
3.2	Empfang der Sms mit <code>gsmlib v1.10</code>	5
3.3	Verarbeitung der SMS mit <code>SmsCommunicator</code>	5
4	Anwendungsbeispiel TippligaHandler	6
4.1	Befehl <code>tipptag</code>	6
4.2	Befehl <code>tippen</code>	6
4.3	Beispiel	6
5	Bewertung/Ausblick	7

1 Motivation: die Tippliga

1.1 Modus der Tippliga

Bei der Tippliga handelt es sich um ein Fußball-Bundesliga - Tippspiel mit einem besonderen Spielmodus. Getippt werden die Toto-Tipps der kompletten Saison. Allerdings werden nicht einfach die richtigen Tipps über die ganze Saison pro Spieler aufsummiert, sondern an jedem Spieltag treten jeweils zwei Tipper gegeneinander an. Die Verhältnis der richtigen Tipps entscheidet dabei über die Vergabe der Punkte des Tippspiels, analog zur Bundesliga erhält der Sieger 3 Punkte, der Verliere 0, oder im Falle eines Unentschiedens erhalten beide Tipper einen Punkt. Um zu gewährleisten, daß alle Tipper gleichoft gegeneinander tippen müssen, ist die Mitspielerzahl auf genau 18 beschränkt, wobei man dann einfach den Bundesliga-Spielplan nach Zuordnung aller Tipper auf jeweils einen Bundesligacub auch als Tippplan benutzen kann. Um am Ende einer Saison einen Sieger zu bestimmen, faßt man abschließend alle Tippspiele analog zur Bundesliga in tabellenform zusammen und hat so die Möglichkeit, den Tippligameister zu küren.

1.2 Logistik

Der Tippligamodus erfordert zum einen, bis zum Tippabgabeschluss des jeweiligen Spieltags die Tipps aller Tipper zu sammeln und allen anderen Tippern zukommen zu lassen. Zum anderen muß nach jedem Spieltag/Spiel der Tipptag und die Tippligatabelle neu ausgewertet werden. Um diese Aufgaben zu automatisieren, wurde mit PHP die Website <http://www.tippliga.info> gebastelt. Möchte ein Tipper seine Tipps über diese Website abgeben, ist es allerdings erforderlich, daß man einen Rechner mit Internetanschluß zur Verfügung hat.

Um eine mobile Alternative zu haben, ohne im Netz zu sein, ist das Projekt "Tippen per SMS" entstanden. Da fast alle Tipper ein Mobiltelefon besitzen, bekommt jeder einen günstigen und ständigen Zugriff auf seine Tipps. Bei der Entwicklung wurde jedoch darauf geachtet, eine Sms-Schnittstelle zu bekommen, auf die beliebige andere Anwendungen aufbauen können. "Tippen per SMS" ist daher lediglich ein Anwendungsbeispiel für den `SmsCommunicator`.

2 Hardware

Die Hardwarevoraussetzungen sind relativ übersichtlich. Zum einen benötigt man natürlich ein Mobiltelefon, um die Sms zu empfangen. Zum anderen braucht man einen Rechner, auf dem die benötigte Software zur Verarbeitung der Sms läuft und an den per Datenkabel das Mobiltelefon angeschlossen wird. Je nach Ausstattung des Mobiltelefons und des Rechners kann die Anbindung auch kabellos per Infrarot oder Bluetooth erfolgen.

3 Software

Die Verarbeitung der Sms erfolgt in zwei Schritten. Im ersten Schritt muß die Sms aus dem Mobiltelefon ausgelesen werden. Dies erledigt die Bibliothek `gsmLib v1.10`. Wenn eine Sms empfangen wurde, wird auf dem Rechner im zweiten Schritt das selbstgeschriebene Java-Programm `SmsCommunicator` gestartet, das die weitere Verarbeitung der Sms übernimmt.

3.1 Konvention des SMS-Formats

Um eine Sms sinnvoll verarbeiten zu können, muß diese mindestens vier Strings in bestimmter Reihenfolge enthalten, die jeweils durch Whitespace voneinander getrennt sein müssen. Jeder zusätzliche String gilt als Parameter und wird in `data` zusammengefaßt:

Name	Funktion
<code>context</code>	definiert die aufzurufende Anwendung
<code>cmd</code>	definiert den auszuführenden Befehl
<code>name</code>	Benutzername
<code>pwd</code>	Passwort
<code>data</code>	optionale Parameter, z.B. Spieltagnr und Tipps

Der erste Parameter ist erforderlich, um mit einem System mehrere verschiedene Anwendungen ansprechen zu können. Name und Passwort sind für Mehrbenutzerbetrieb unerlässlich. Außerdem kann durch die Benutzerauthentifizierung verhindert werden, daß durch böswillige Benutzung Antwort-Sms erzeugt werden, die in entsprechenden Mengen einige Kosten verursachen könnten.

3.2 Empfang der Sms mit `gsmlib v1.10`

`gsmlib v1.10` ist eine Linux-Bibliothek, mit der man mit Mobiltelefonen kommunizieren kann. Dabei erfolgt der Zugriff über das eingebaute GSM-Modem des Mobiltelefons. Dadurch wird erreicht, dass man mit Standardbefehlen auf Mobiltelefone verschiedener Hersteller zugreifen kann. Mit den verschiedenen Kommandozeilentools kann man beispielsweise das Telefonbuch oder gespeicherte Sms auslesen und bearbeiten oder auch Sms verschicken.

Das für dieses Projekt verwendete Tool ist `gsmsmsd`, ein Dämon, der zwei Funktionen zur Verfügung stellt. Zum einen fragt er laufend das Mobiltelefon auf eingegangene Sms ab. Im Erfolgsfall wird die Sms ausgelesen und über den Standarteingabestrom an ein externes Programm weitergegeben, welches `gsmsmsd` startet. Dieses gibt man beim Start von `gsmsmsd` an. Zum anderen kann `gsmsmsd` auch Sms verschicken, indem man eine Datei mit Empfängernummer und Textnachricht in ein Spoolverzeichnis schreibt.

3.3 Verarbeitung der SMS mit `SmsCommunicator`

Wie oben beschrieben, wird eine empfangene Sms von `gsmsmsd` ausgelesen. `gsmsmsd` startet dann den `SmsCommunicator`. Dieser liest zu Beginn den Eingabestrom aus und ereugt daraus ein Sms-Objekt. Dieses enthält nur die Bestandteile Sendernummer und Text als Strings. Der Text wird anschließend in seine zusammenhängenden Strings zerlegt, um daraus ein Message-Objekt zu erzeugen. Das Message-Objekt enthält einen String `context`, um verschiedene Anwendungen ansprechen zu können. Diese werden in Unterklassen von der Klasse `ContextHandler` gekapselt. Auf diese Weise lassen sich die einzelnen Anwendungen gut trennen. Der `context` entsprechende `ContextHandler` wird dynamisch geladen und instanziiert.

Jeder `ContextHandler` enthält eine Authentifizierungsmethode und Methoden um die jeweiligen Befehle zu verarbeiten. Kann der in der Sms enthaltene Benutzer erfolgreich authentifiziert werden, wird anschließend der geforderte Befehl mit evtl. Parametern abgearbeitet. Da der ausführende Rechner im Netz steht, kann das Programm hierfür auf eine ganze Reihe zusätzlicher Ressourcen wie z.B. Datenbanken oder Mailserver zugreifen. Dann wird eine Antwort-Sms generiert, die Auskunft über den Erfolg des Befehls geben oder auch als Vorlage für eine neue Sms an den `SmsCommunicator` dienen kann. Die Sms wird zum Versenden ins Spoolverzeichnis des `gsmsmsd` geschrieben, bevor der `SmsCommunicator` terminiert.

4 Anwendungsbeispiel TippligaHandler

Der `TippligaHandler` greift auf die Datenbank der Website `tippliga.info` zu. So können sich die Tipper mit ihrem von der Website bekannten Login authentifizieren. Zur Zeit verarbeitet der `TippligaHandler` zwei Befehle:

4.1 Befehl `tipptag`

Der Befehl `tipptag` wird aufgerufen mit der zu tippenden Spieltagnummer als Parameter. Als Antwort-Sms wird eine Sms generiert, die als Vorlage für den zweiten Befehl dient. Sie enthält bereits alle benötigten Stings, den `context` Tippliga, den Befehl `tippfen`, Name und Passwort des Tippers und eine Spieltagübersicht, in die der Tipper einzig nur noch seine Tipps eintragen muß.

4.2 Befehl `tippfen`

Der Befehl `tippfen` erwartet in den Parametern die Spieltagnummer und die neun Tipps, um sie für den entsprechenden Tipper in die Datenbank eintragen zu können. die generierte Antwort-Sms enthält lediglich eine kurze Bestätigung, daß die Tipps erfolgreich eingetragen wurden.

4.3 Beispiel

Folgende Sms fordert für den 31. Spieltag die Spieltagübersicht an:

```
tippliga tipptag pfox *** 31
```

Der `SmsCommunicator` erzeugt folgende Antwort-Sms und schickt diese an den Absender zurück:

```
tippliga tippfen pfox *** 31 SVW:BSC_  
BVB:WOB_ FCN:60M_ COT:HSV_ LEV:BIE_  
BOC:MGL_ ROS:S04_ H96:VFB_ FCB:FCK_
```

Dieser kann die Sms mit seinen Tipps füllen und wiederum zurückschicken:

```
tippliga tippen pfox *** 31 SVW:BSC_1
BVB:WOB_0 FCN:60M_2 COT:HSV_0 LEV:BIE_1
BOC:MGL_0 ROS:S04_1 H96:VFB_2 FCB:FCK_1
```

Ist das Eintragen der Tipps erfolgreich, erhält der Absender eine Bestätigungs-Sms vom `SmsCommunicator`:

```
Die Tipps wurden erfolgreich eingetragen
```

5 Bewertung/Ausblick

Man könnte für die Tippliga noch weitere Befehle implementieren, um sich z.B. die Ergebnisse des Spieltags oder die Tippliga-Tabelle zuschicken zu lassen. Allerdings stieße man schnell an die Grenze, daß eine Sms nur 160 Zeichen enthalten kann. Die Grenze könnte man Richtung Nutzer aufheben, indem man einfach mehrere Sms verschickt. In die andere Richtung wäre das technisch nur schwer möglich, da man zusammengehörende Mails nicht zuordnen könnte.

Als weitere Anwendung könnte man beispielsweise einen `MailHandler` entwickeln, der einen Mailserver anspricht, um sich über den Status seiner Mailbox zu informieren oder sich einzelne Mails zuschicken zu lassen.

Im Grunde kann man alle Geräte, die sich an einen Rechner anschließen lassen, über einen entsprechenden Handler per Sms fernsteuern. Beispielsweise ließe sich so ein Türöffner, der an einer Telefonanlage angeschlossen ist, über Modem per Sms öffnen. Allerdings muß man sich dabei bewußt sein, daß jede Sms Kosten verursacht, was weitere umfangreichere Anwendungen für den `SmsCommunicator` sehr einschränkt.

Im Umfang eingeschränkt wird der `SmsCommunicator` auch durch die Geschwindigkeit, mit der Sms empfangen und verschickt werden. In Tests ließen sich nicht mehr als 6 Sms pro Minute verschicken. Ein Massenversand ist damit praktisch ausgeschlossen.

Für das vorliegende Beispiel "Tippen per Sms" ist der `SmsCommunicator` jedoch ausreichend und läuft zuverlässig.