

Bachelorarbeit

Observed Assessment

Übersicht

1. Einleitung

- Motivation
- Aufgabenstellung

2. Verwendete Software

- Das Assessment Center
- Grundlagen und Methoden der Mimikerkennung
- SIPBILD - Das Projekt
- Einsatz des Multimodeltracker

3. Softwaretechnische Konzeption

4. Inhaltliche Konzeption und Demonstration

Übersicht

1. Einleitung

2. Verwendete Software

3. Softwaretechnische Konzeption

- Kommunikation mit XML-RPC
- Verwendete Swing-Konzepte
- Programmfluss
- Erweiterung der Oberfläche

4. Inhaltliche Konzeption und Demonstration

Motivation

- **Kommunikation mit dem PC**
 - Bedienung in den allermeisten Fällen mit den Eingabegeräten Maus und Tastatur
 - Sehr anders als sämtliche „normale“ Kommunikation des Menschen
- **Neue Möglichkeiten**
 - Leistungsfähigkeit moderner Computer ermöglicht auch Beachtung anderer Eingabekanäle
 - Beispiel: Eingabe längerer Texte mit Hilfe von Spracherkennung

Motivation

- **Zwischenmenschliche Kommunikation**
 - Findet über viele verschiedene Kanäle gleichzeitig statt
 - Untersuchung: Verbaler Anteil an der zwischenmenschlichen Kommunikation macht nur 20% aus
- **Nonverbale Kommunikation**
 - Drückt sich nicht nach den konventionalisierten Regeln einer Sprache aus
 - Teilweise bewußt, teilweise unbewußt
 - Beispiele: Körperhaltung, Geruch, Gestik und Mimik

Motivation

- Mimik
 - Sichtbare Bewegung der Gesichtsoberfläche
 - Träger der Emotionen des Menschen
 - Schon im Kindesalter lernt man das Erkennen und Ausüben verschiedener Mimiken
- Idee:
 - Automatische Analyse der Mimik durch Computer
 - Rückschluß von Mimik auf Emotionen
 - Software reagiert auf die erkannte Mimik

Aufgabenstellung

- Modifikation des Assessment-Laufzeitsystems
 - Beobachtung der Mimik des Benutzers während der Bearbeitung eines Assessments
 - Analyse der Bilder mit Mimikerkennungs-Software des FORSIP-Teilprojekt SIPBILD
- Gesten und Mimik des Benutzers sollen zu unmittelbarer Reaktion des Programms führen

Übersicht

1. Einleitung

2. Verwendete Software

- Das Assessment Center
- Grundlagen und Methoden der Mimikerkennung
- SIPBILD - Das Projekt
- Einsatz des Multimodeltracker

3. Softwaretechnische Konzeption

4. Inhaltliche Konzeption und Demonstration

Das Assessment Center

- Das Projekt
 - Entwickelt am Passauer Institut für Informationssysteme und Softwaretechnik (IFIS)
 - E-Learning-Plattform für Erstellung, Durchführung und Auswertung interaktiver Übungen und Prüfungen
 - Inhalte darstellungunabhängig in XML kodiert

Das Assessment Center

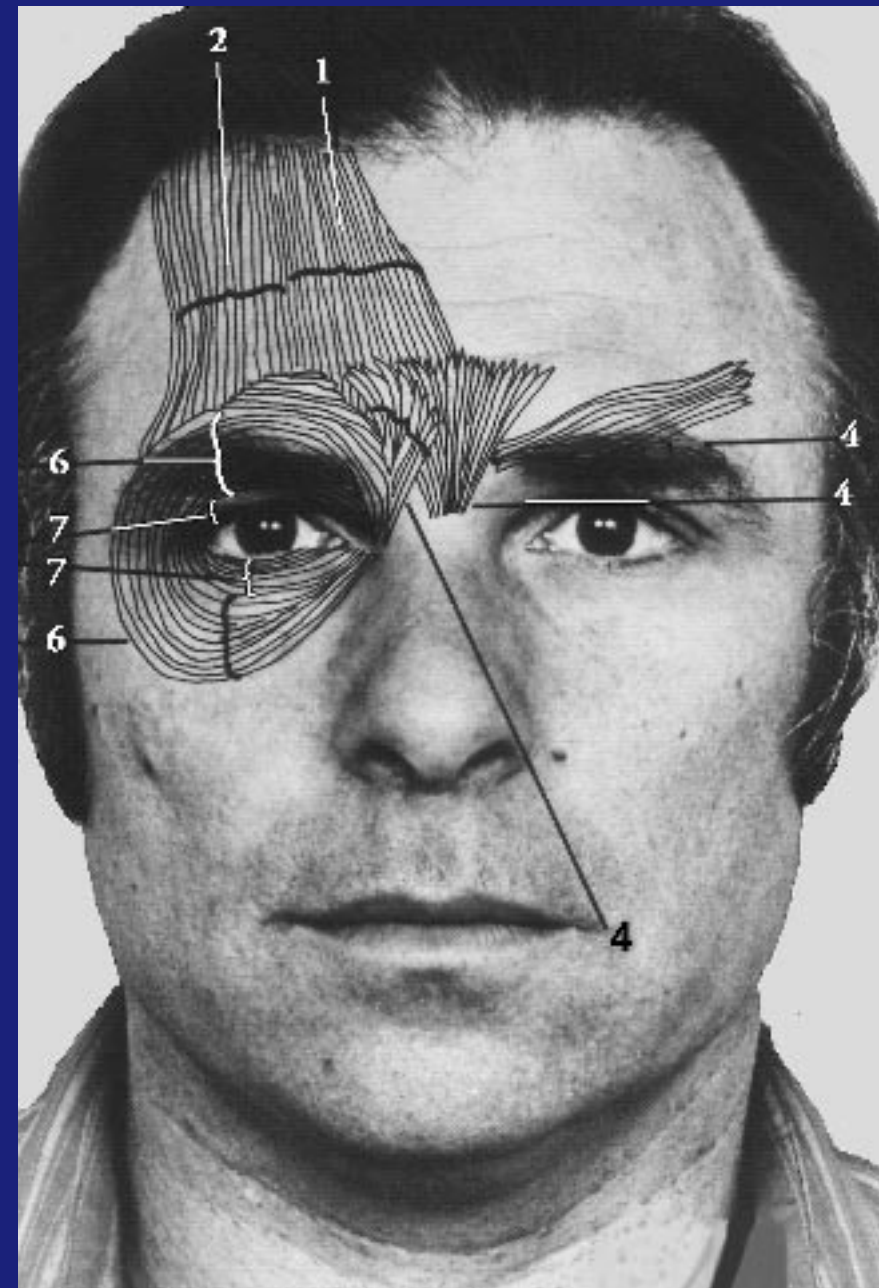
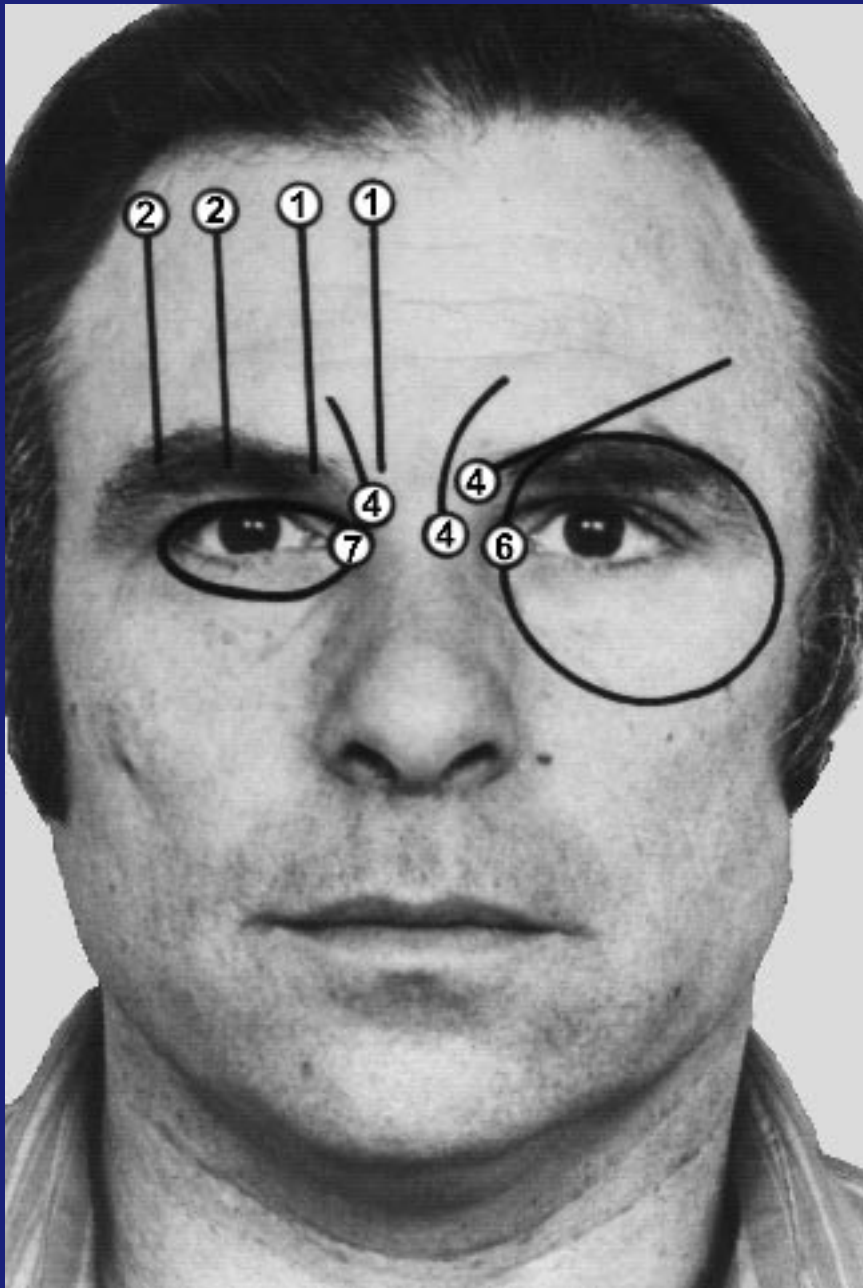
- Java-Client
 - Entwickelt für Prüfungsmodus
 - Programmablauf besser kontrollierbar als bei Ablauf im Webbrowser
 - Fragen und Antworten in zufälliger Reihenfolge
 - Timer überwacht die Bearbeitungszeit
- Implementierung
 - Oberfläche in Java Swing umgesetzt
 - Model-View-Controller-Architektur
 - Verwendet für Inhalte ebenfalls XML-Format
 - Konfiguration über Resource Bundle

Mimikerkennung: Grundlagen

- Mimik des Menschen
 - Sichtbare Bewegung der Gesichtsoberfläche
 - Besonders geprägt durch Mund und Augen
 - Erzeugt durch die Mimische Muskulatur
- Interpretation der Mimik
 - Auch von Menschen sehr oft falsch eingeschätzt
 - Für automatisierte Interpretation ist objektives Klassifizierungssystem nötig

Mimikerkennung: Grundlagen

- FACS: Facial Action Coding System
 - Entwickelt im Jahr 1974 von den Psychologen Paul Ekman und Wallace Friesen
 - Grundlage: 44 Action Units
 - Definiert auf anatomischer Basis
 - Zusammenspiel der Action Units kodiert mimisches Erscheinungsbild



Schema: Action Units

Beispiel: Action Unit 1

- Heben der inneren Augenbraue
„In der Kopfhaut und der Stirn hebt ein großer Muskel die Augenbrauen. Dieser Muskel verläuft vertikal vom oberen Teil des Kopfes zu den Augenbrauen und umfasst praktisch die gesamte Stirn.“
- Sichtbare Veränderungen:
 - Innerer Teil der Augenbrauen wird nach oben gezogen
 - Haut in Stirnmitte bildet horizontale oder gekrümmte Falten
 - Außenspitze der Augenbrauen bewegt sich evtl. etwas zur Mitte



Action Unit 1

Mimikerkennung: Grundlagen

- Ablauf der automatisierten Mimikerkennung:
 1. Detektion der genauen Position des Kopfes
 2. Extraktion der Gesichtszüge
 3. Klassifikation des Gesichtsausdrucks
- Zeitabhängige Analyse
 - Genauere Klassifikation möglich durch Betrachten der Veränderung der Gesichtszüge über Zeitraum hinweg
 - Ermöglicht zudem die Detektion von Kopfgesten wie Nicken und Kopfschütteln

SIPBILD - Das Projekt

- SIPBILD
 - „Mimik- und Gestikerkennung in Videobildfolgen“
 - FORSIP-Projekt
 - Lehrstuhl Informatik IX, TU München
- SIPBILD: Mimikerkennung
 - Entwickelt von Matthias Wimmer
 - sucht in Kamerabild nach menschlichem Gesicht
 - extrahiert und klassifiziert dessen Züge

SIPBILD - Das Projekt

- Eigenschaften
 - Läuft autonom, keine menschliche Interaktion nötig
 - Analysiert Kamerabild in Echtzeit
 - robust gegen Brille, Bärte etc.
 - robust gegen Beleuchtungsänderungen
 - robust gegen Bewegungen im Hinter- und Vordergrund
- keine Laborbedingungen nötig, im realen Betrieb einsetzbar

SIPBILD - Das Projekt

- **Gesichtsmodell**
 - Aus Integration mehrerer hundert Bilder verschiedenster Menschen
 - Unterschiedliches Alter, Geschlecht, Hautfarbe etc.
 - Jeweils 134 definierte Gesichtspunkte markiert
- Point Distribution Model



Point Distribution Model

SIPBILD - Das Projekt

- Einpassvorgang von Gesicht in Modell
 - Kombination zweier Techniken:
 - Kantendetektion/-extraktion
 - Analyse des Grauwertprofils
 - erhöhte Detektionsgenauigkeit
- Zusätzlich: Tracking-Algorithmus
 - Ermöglicht Verfolgung des detektierten Gesichts über mehrere Bilder hinweg

SIPBILD - Das Projekt

- **Klassifikation der Mimik**
 - Nachteile des Modell von Ekman und Friesen:
 - Nicht für Anwendung in Software entwickelt
 - Basiert auf dem statischen Gesichtsausdruck
 - Entwicklung eines eigenen Modells
- **Vorgehensweise**
 - Mimiken repräsentiert durch 4 Positionsparameter und 17 Deformationsarten
 - Einteilung der betrachteten Zeitperiode in 23 Schritte
 - 483-stelliger mimic feature vector
 - Klassifikation mit binärem Entscheidungsbaum

SIPBILD - Das Projekt

- Bisher unterstützte Mimiken:
 - neutral
 - lachend
 - überrascht
- Unterstützte Kopfgesten:
 - Nicken
 - Kopfschütteln

Demo: Multimodeltracker

SIPBILD - Das Projekt

- Implementierung
 - Programmierung in C++
 - Betriebssystem Microsoft Windows
 - verwendet Open CV-Bibliothek von Intel Corporation
 - enthält Algorithmen für Human-Computer Interaction, Object Identification, Face Recognition etc.
 - Kommunikation mit anderen Anwendungen möglich
 - integrierter XML-RPC-Client
 - Übermittlung der erkannten Emotionen als Integerwerte

Einsatz des Multimodeltracker

- Einrichtung und Aufruf
 - Keine Installation nötig
 - Konfiguration über Kommandozeilenparameter
 - Detektionseinstellungen (Yes/No, Emotion etc.)
 - Zeitkonstanten (Zeit für initialen Einpassvorgang etc.)
 - Daten des anzusprechenden XML-RPC-Server
- Programmablauf
 - Autonome Arbeitsweise
 - einige Steuerungsmöglichkeiten (Reinitialisieren etc.)

Übersicht

1. Einleitung
2. Verwendete Software
- 3. Softwaretechnische Konzeption**
 - Kommunikation mit XML-RPC
 - Verwendete Konzepte
 - Programmfluss
 - Erweiterung der Oberfläche
4. Inhaltliche Konzeption und Demonstration

Kommunikation mit XML-RPC

- Remote Procedure Calls
 - Funktionsaufrufe zwischen verschiedenen Anwendungen
 - Übermittlung von Parametern und Rückgabewerten
 - Über Netzwerke/Internet möglich
- XML-RPC
 - 1998 von Dave Winer für Blogsoftware spezifiziert
 - Implementierungen für zahlreiche Betriebssysteme und Programmiersprachen vorhanden
 - Java-Implementierung von Apache Software Foundation

Kommunikation mit XML-RPC

- XML-RPC: Funktionsweise
 - XML-RPC Nachricht ist eine HTTP-POST Anfrage
 - Aufzurufende Funktion und Parameter in XML kodiert
 - Übermittlung des Rückgabewerts analog
 - Typen:
 - boolean, integer, double, string
 - Binärdaten in Base64-Encoding
 - zusätzlich Datenstrukturen definiert
 - Auch Übermittlung von Fehlermeldungen möglich

Kommunikation mit XML-RPC

- **Multimodeltracker**
 - Ruft bei Detektion einer Mimik die Funktion `newMimicDetected` auf
 - Verläßt/Betritt der Benutzer Sichtbereich der Kamera: `personWentAway` bzw. `newPersonDetected`
- **Observed Assessment Center**
 - Integrierter XML-RPC-Server, als Thread gestartet
 - Muß die angerufenen Funktionen implementieren
 - Gibt die erkannten Mimiken an Assessment Center weiter

Mimik-Signal: Probleme

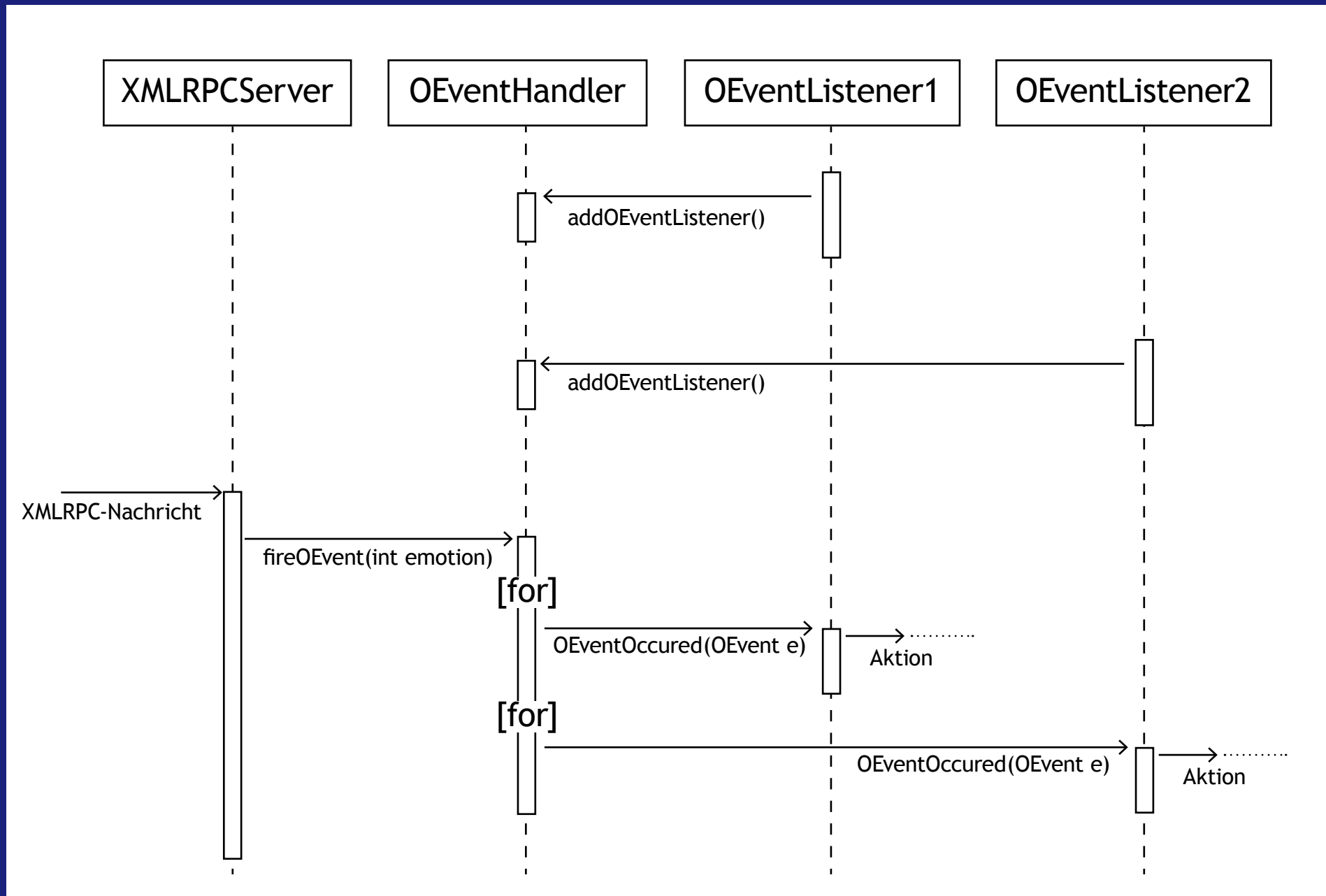
- Schnell wechselndes Signal
 - Führt zu unkontrollierter Assessment-Steuerung
 - Lösung: Signaldämpfung zur Stabilisierung
 - neues Signal nur jede Sekunde akzeptiert
 - ansonsten Ignorieren des Signals
- Langer Detektionszeitraum
 - Nach Mimikdetektion muß Software vor erneuter Detektion erst in wieder in Neutral-Modus wechseln
 - Lösungsansatz: Aufsplitten längerer statischer Mimik
 - Aber: Führt zu unkontrolliertem Verhalten

Verwendete Konzepte

- Singleton-Klassen
 - Beispiel: ObservedEventHandler
 - Konstruktor ist geschützt
 - Objekt wird nur über statische Zugriffsfunktion nach außen gegeben: getInstance()

Verwendete Konzepte

- Delegation Model
 - An Mimik interessierte Klassen (ObservedEventListener) melden sich beim zentralen ObservedEventHandler an
 - XML-RPC-Server erzeugt bei eingehendem Signal ein ObservedEvent und ruft den ObservedEventHandler auf
 - ObservedEventHandler delegiert den Aufruf an alle bei ihm registrierten Listener
 - Zentrale Ansprechstelle für Mimiksignale
 - Alle Klassen können selbst entscheiden, ob und wann sie benachrichtigt werden wollen



Grafik: Programmfluß

Erweiterung der Oberfläche

- **Mimikanzeige**
 - Benutzer hat jederzeit Einblick in die gerade von der Mimikerkennung erkannte Emotion
- **Demonstrations-Modus**
 - Nach Inaktivitätszeit automatisch Neustart des Assessments
 - Frei konfigurierbar über `observed.properties`



Mimikanzeige

Erweiterung der Oberfläche

- **Auswertungsbildschirm**
 - Auflistung über richtig und falsch beantwortete Fragen
 - Gibt Benutzer Überblick über erzielt Resultat
 - Nach eigenem Timeout erfolgt wieder Neustart des Assessment Center

Test erfolgreich abgegeben.

Ihr Ergebnis:

Richtig beantwortete Fragen: 5

Falsch beantwortete Fragen: 5

Auswertungsbildschirm

Demo: Observed Assessment