

ATEO-SYSTEM

KOMPONENTE: SAMS

SOFTWARE-SPEZIFIKATION

| | |
|----------------|--|
| Version | 2.3.3 |
| Datum | 15.10.2010 |
| Autoren | Prof. Dr. Klaus Bothe Michael Hildebrandt Nicolas Niestroj |

INHALT

| | | |
|-------|---|----|
| 1 | Zweck und Aufbau des Dokuments..... | 4 |
| 2 | Zielbestimmung..... | 4 |
| 2.1 | Muskriterien..... | 4 |
| 2.2 | Wunschkriterien | 5 |
| 2.3 | Abgrenzungskriterien | 5 |
| 3 | Produkteinsatz | 5 |
| 3.1 | Anwendungsbereiche | 5 |
| 3.1.1 | Forschung | 5 |
| 3.1.2 | Lehre..... | 7 |
| 3.1.3 | Art des Einsatzes | 7 |
| 3.2 | Zielgruppen..... | 8 |
| 4 | Beschreibung der Anwendungsfälle..... | 8 |
| 4.1 | Anwendungsfälle im Überblick..... | 8 |
| 4.2 | Tracking durchführen | 9 |
| 4.2.1 | Überblick | 9 |
| 4.2.2 | Eingaberaum der Steuereingaben..... | 10 |
| 4.3 | Tracking überwachen und führen..... | 12 |
| 4.3.1 | Überblick | 12 |
| 4.3.2 | Zugriff auf die Simulationsdaten | 12 |
| 4.3.3 | Eingriffe in den Tracking-Prozess | 13 |
| 4.3.4 | Fehlbedienung..... | 15 |
| 4.4 | Versuch konfigurieren | 15 |
| 4.4.1 | Überblick | 15 |
| 4.4.2 | Funktionale Beschreibung..... | 15 |
| 4.4.3 | Beschreibung der Daten..... | 17 |
| 4.5 | Versuchsbeschreibung entwerfen und einlesen | 18 |
| 4.5.1 | Überblick | 18 |
| 4.5.2 | Streckenbeschreibung..... | 18 |

| | | |
|-------|---|----|
| 4.5.3 | Versuchsbeschreibung | 19 |
| 4.6 | Versuchsdaten protokollieren | 20 |
| 4.6.1 | Überblick | 20 |
| 4.6.2 | Funktionale Beschreibung | 20 |
| 4.6.3 | Beschreibung der Daten | 21 |
| 5 | Nichtfunktionale Anforderungen & Qualitätsziele | 26 |
| 5.1 | Übersicht | 26 |
| 5.2 | Zuverlässigkeit / Effizienz | 26 |
| 5.3 | Bedienbarkeit | 27 |
| 5.4 | Wartbarkeit | 27 |
| 5.5 | Variabilität | 27 |
| 6 | Technische Systemumgebung | 27 |
| 6.1 | Referenzsystem | 27 |
| 6.1.1 | Hardware | 27 |
| 6.1.2 | Software | 28 |
| 6.2 | Produkt-Schnittstellen | 28 |
| 6.2.1 | Das ATEO-System | 28 |
| 6.2.2 | Versuchsaufbau | 29 |
| 7 | Literatur | 30 |
| 8 | Quellen | 31 |
| 9 | Glossar | 32 |
| 10 | Anhang: Implementationsdetails | 34 |
| 10.1 | Streckenanimation | 34 |
| 11 | Abweichungen und Fehler | 34 |
| 11.1 | Versuchssteuerung | 34 |

1 ZWECK UND AUFBAU DES DOKUMENTS

Dieses Dokument beschreibt die Anforderungen an die Software-Komponente SAMs, welche einen Teil des ATEO-Systems darstellt. Es wurde im Zuge eines Reverse Engineerings gewonnen und beschreibt daher den Ist-Zustand eines existierenden und benutzten Systems in Form von Anforderungen.

Die Struktur dieses Dokuments *orientiert* sich hierbei an der in [2] vorgeschlagenen Struktur für ein Pflichtenheft. Hierbei wird in *Abschnitt 2* zunächst eine Zielbestimmung durchgeführt, welche die Muss-, Wunsch- und Abgrenzungskriterien von SAMs erläutert. Daraufhin erfolgt in *Abschnitt 3* die Darstellung des Produkteinsatzes, welche die Anwendungsbereiche und die jeweiligen Zielgruppen vorstellt. In *Abschnitt 4* werden die Anwendungsfälle in der Übersicht und auch im Detail beschrieben. Danach erfolgt eine Darstellung der nichtfunktionalen Anforderungen, welche an SAMs gestellt werden (*Abschnitt 0*). Die technische Versuchsumgebung wird dann in *Abschnitt 6* vorgestellt. In *Abschnitt 7* wird die in diesem Dokument verwendete Literatur angegeben. Neben der Literatur gibt es weitere Quellen, welche aktuelle Informationen zum Projekt ATEO enthalten. Diese werden in *Abschnitt 8* vorgestellt. Das Glossar, welches die Bedeutung der hier verwendeten Begriffe erläutert, ist in *Abschnitt 9* angegeben. Schließlich finden sich im Anhang noch Informationen zu Implementationsdetails und zu Fehlern und Abweichungen zwischen der Implementation und der Verhaltensspezifikation (*Abschnitte 10 und 11*).

2 ZIELBESTIMMUNG

Das ATEO-System soll es den Anwendern ermöglichen rechnergestützte psychologische Experimente durchzuführen, auf dessen Basis Untersuchungen zur Funktionsteilung zwischen Mensch und Maschine durchgeführt werden, um damit wiederum Rückschlüsse auf die Arbeitsteilung zwischen Entwicklern und Operateuren ziehen zu können (siehe hierzu [3], [4] und [5]).

SAMs soll dabei als Komponente des ATEO-Systems die Simulation eines dynamischen Prozesses in Form einer Tracking-Aufgabe implementieren. Zudem soll es anderen Komponenten des ATEO-Systems dabei möglich sein, den Prozess zu überwachen und diesen auch zu beeinflussen.

2.1 MUSSKRITERIEN

Es gilt dabei die folgenden Anforderungen umzusetzen:

- **Tracking**
 - Eine Tracking-Aufgabe kann sowohl durch eine einzelne Versuchsperson als auch kooperativ durch zwei Versuchspersonen durchgeführt werden.
 - Als Eingabegerät für die Versuchspersonen können Joysticks verwendet werden.
 - Es soll anderen Software-Komponenten ermöglicht werden die Daten des laufenden Tracking-Prozesses abzufragen und auf diesen einzuwirken.
 - Die Tracking-Aufgabe kann durch die Versuchspersonen gestartet werden.
- **Logfile**
 - Während der Durchführung der Tracking-Aufgabe sollen relevante Versuchsdaten in Form einer CSV-Datei protokolliert werden.
 - Der zeitliche Abstand zwischen zwei aufeinanderfolgenden Logfile-Einträgen soll zwischen 30 und 49ms liegen.
- **Versuch**

- Die Versuchsparameter sollen vor der Versuchsdurchführung durch den Versuchsleiter mittels eines Dialogs konfigurierbar sein.
- Es kann der gesamte Versuch durch externe Text- und Grafikdateien konfiguriert werden. Hierbei kann jeder einzelne Versuchsschritt, jede dazugehörige Strecke sowie der gesamte Versuchsablauf festgelegt werden. Diese Dateien sollen eingelesen und die entsprechende Strecken sowie der entsprechende Versuchsablauf durch SAMs dargestellt bzw. ausgeführt werden.
- Der Versuchsleiter kann den Versuch starten und abbrechen.

2.2 WUNSCHKRITERIEN

- Das Durchführen des Trackings soll auch auf leistungsschwächeren Computersystemen (z.B. Laptops, Netbooks) ermöglicht werden, ohne dass der zeitliche Abstand zwischen zwei Log-Datensätzen außerhalb des Zeitintervalls von [30, 48) ms liegt.
- Es sollen unterschiedliche Steuergeräte (Joysticks, Lenkräder, ...) als Eingabegeräte für die Versuchspersonen, welche die Tracking-Aufgabe absolvieren, unterstützt werden.
- Das Erscheinungsbild der Tracking-Simulation soll variabel konfigurierbar sein, so dass beispielsweise Position, Form, Farbe und Größe der Objekte auf der Strecke als auch die Strecke selber konfiguriert werden können.
- Die physiologische Beanspruchung der Versuchspersonen, welche die Tracking-Aufgabe durchführen, soll während der Durchführung der Tracking-Aufgabe erfasst und geloggt werden.

2.3 ABGRENZUNGSKRITERIEN

Die folgenden Kriterien sollen von SAMs *nicht* umgesetzt werden:

- Die Vergabe der Ziele, welche die Versuchspersonen während des Trackings zu erreichen versuchen.
- Die Auswertung des während der Versuchsdurchführung erstellten Logfiles.
- Die Funktionalitäten für die Simulation von realen Automaten und Assistenzsystemen.
- Die Funktionalität eines umfangreichen und komfortablen Operateursarbeitsplatzes.

3 PRODUKTEINSATZ

3.1 ANWENDUNGSBEREICHE

3.1.1 FORSCHUNG

Das Produkt wird im Rahmen des Projektes *Arbeitsteilung Entwickler-Operateur* (ATEO) entwickelt und eingesetzt, welches sich mit der asynchronen Arbeitsteilung zwischen Entwicklern und Designern sowie Operateuren und Anwendern eines Mensch-Maschine-Systems beschäftigt (siehe *Abbildung 1*, [3]). Erstere planen, konzipieren und implementieren dieses System, während letztere das implementierte System schließlich benutzen.

Die Erkenntnisgewinnung im ATEO-Projekt erfolgt durch Experimente, welche rechnergestützt durchgeführt werden. Im Zentrum steht hierbei das Konzept der *Socially Augmented Microworld* (SAM) (siehe hierzu [4], [5]). Hierbei wird in SAM ein dynamischer Prozess in Form einer Tracking-Aufgabe modelliert, welcher zu einem gewissen Grad unvorhersehbar ist, aber dennoch in der Retrospektive erklärt werden kann. Dies wird dadurch erreicht, dass SAM einen sozialen Faktor enthält, nämlich die *Mikroweltbewohner* (MWB). Bei den MWB han-

delt es sich um ein bis zwei Versuchspersonen, welche die Aufgabe haben ein Tracking-Objekt auf einer vorgegebenen Strecke entlang zu steuern (vergleiche *Abbildung 2*).

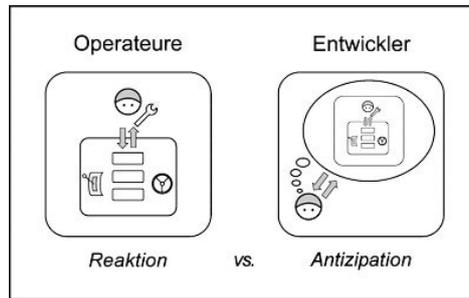


Abbildung 1 - Die Arbeitsteilung zwischen Entwicklern und Operateuren (siehe [3]).

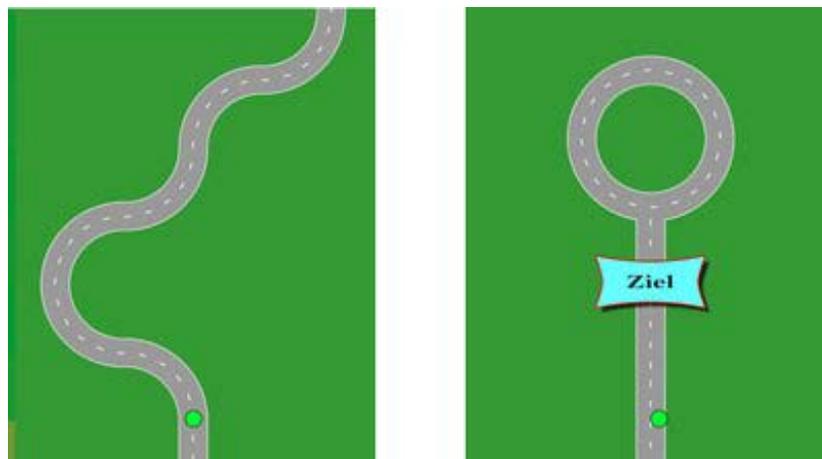


Abbildung 2 - Die Tracking-Aufgabe von SAM (siehe [3]).

Dies kann kooperativ oder nur durch einen der MWB durchgeführt werden. Hierbei kann die Tracking-Aufgabe unter verschiedenen Zielstellungen durchgeführt werden, welche von der aktuellen Fragestellung des Versuchs abhängen. Beispielsweise kann ein Ziel lauten möglichst schnell das Tracking durchzuführen oder aber das Tracking-Objekt möglichst genau entlang der Spur zu steuern. Die Zielvergabe geschieht dabei außerhalb von SAMs mittels zusätzlicher Laptops.

Eine weitere Versuchsperson wird als Operateur tätig, welcher das Tracking der MWB überwacht und Eingriffe in den Steuerungsprozess tätigt, falls dies nötig sein sollte. Anstatt einem menschlichen Operateur können auch **Automatiken** (AM) für die Prozessüberwachung und -führung eingesetzt werden. Der Operateur bzw. die Automatik jedoch gehören selbst nicht mehr zu SAM. *Abbildung 3* verdeutlicht diesen Zusammenhang grafisch.

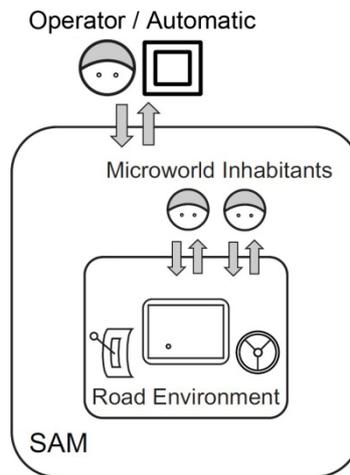


Abbildung 3 - Der Zusammenhang von SAM, den MWB sowie dem Operator bzw. den Automaten (nach [3]).

Dieses Konzept von SAM wurde u.a. in Smalltalk umgesetzt. Die entsprechende Software wird als *Socially Augmented Microworld in Smalltalk* (SAMs) bezeichnet und ist Gegenstand dieses Dokuments.

3.1.2 LEHRE

Dieses Produkt wird zudem als Fallstudie für die Entwicklung einer Software in einem interdisziplinären Team herangezogen werden, welche es den Studenten ermöglicht Einblicke in die Vorgehensweisen und Probleme einer solchen Software-Entwicklung zu gewinnen. Darüber hinaus können die Studenten durch die praktische Anwendung von softwaretechnischen Methoden ihr Wissen bezüglich des Software Engineerings vertiefen und festigen.

3.1.3 ART DES EINSATZES

VERSUCHSABLAUF IM ÜBERBLICK

Der Versuchsablauf kann in drei Phasen aufgeteilt werden. Die erste Phase ist die Versuchsplanung und -vorbereitung durch den Versuchsleiter. In dieser Phase werden mehrere Vorbereitungen getroffen. Es werden die Haupt- und Nebenaufgaben, welche von den Versuchspersonen auszuführen sind, ermittelt. Weiterhin gilt es den Versuchsablauf zu planen. Ein Versuch besteht aus mehreren Schritten, wobei für jeden einzelnen Schritt eine Strecke erstellt wird und zudem festgelegt werden muss, welcher MWB das Steuern übernimmt und ob es sich um ein kooperatives oder Einzel-Tracking handelt (vgl. *Abschnitt 4.5*).

In der zweiten Phase werden vor dem Beginn des Experiments durch den Versuchsleiter noch einige Parameter und versuchsrelevante Daten (siehe *Abschnitt 4.4*) festgelegt bzw. in SAMs eingegeben. Diese Daten erscheinen dann später auch in der jeweiligen Protokolldatei und dienen der Zuordnung der während der Versuchsdurchführung ermittelten Daten zu den einzelnen Versuchspersonen. Für die Eingabe dieser Daten steht ein einfaches Menü zur Verfügung, welches zudem den Start des Versuchs ermöglicht.

Die Durchführung des Versuchs ist die dritte Phase, in welcher nun die Probanden aktiv werden. Hierzu gehören einerseits die MWB, welche das Tracking-Objekt in mehreren Versuchsschritten kooperativ oder aber einzeln auf verschiedenen Strecken entlang steuern (vgl. *Abschnitt 4.2*) und andererseits der Operateur, welcher das Tracking überwacht und ggf. unterstützend eingreift. Der Operateur kann hierbei weiche Eingriffe oder harte Eingriffe vornehmen. Bei ersteren handelt es sich um das Geben von auditiven und visuellen Hinweisen

und bei letzteren um Eingriffe in die Steuerung des Tracking-Objekts, wie z.B. die Manipulation der möglichen maximalen Geschwindigkeit (vgl. *Abschnitt 4.3.3*).

Der Versuch endet, wenn alle für diesen Versuch vorgesehenen Strecken bzw. Versuchsschritte durch die MWB absolviert wurden. Für den Versuchsleiter liegen nun für jeden einzelnen Schritt des Versuchs Protokolldateien mit den entsprechenden Simulationsdaten vor.

3.2 ZIELGRUPPEN

Für SAMs gibt es drei verschiedene Nutzergruppen: die MWB, den Operateur und den Versuchsleiter.

Während des Versuchs wird das Tracking-Objekt durch die MWB unter der Verwendung von Joysticks über die verschiedenen Strecken gesteuert. Die geforderten Fähigkeiten im Umgang mit den Joysticks hängen hierbei von der aktuellen Fragestellung des Versuchs ab und werden in Fragebögen und vorbereitenden Tests überprüft.

Der Operateur benutzt SAMs nur mittelbar, indem er das Tracking unter Verwendung des *ATEO Master Display* (AMD) überwacht und führt. Dennoch stellt er auch eine Zielgruppe dar, da die Möglichkeiten des AMD bezüglich der Überwachung und des Eingreifens davon abhängig sind, inwiefern SAMs die entsprechenden Daten bereitstellt bzw. die entsprechenden Eingriffe ermöglicht (vgl. *Abschnitt 4.3*).

Der Versuchsleiter konzipiert und konfiguriert den Versuch mit seinen einzelnen Schritten und den dazugehörigen Strecken. Dies erfolgt über externe Text- und Grafikdateien. Darüber hinaus stellt er einige Parameter über das Versuchsleiter-Menü (siehe *Abbildung 8*) unmittelbar vor der Versuchsdurchführung ein. Während der Versuchsdurchführung kann er Einfluss auf den Ablauf der Versuchsdurchführung nehmen.

4 BESCHREIBUNG DER ANWENDUNGSFÄLLE

4.1 ANWENDUNGSFÄLLE IM ÜBERBLICK

Die Anwendungsfälle von SAMs sind in *Abbildung 4* in einem Anwendungsfalldiagramm dargestellt. In einem solchen Diagramm werden die Anwendungsfälle als Ovale dargestellt, Akteure als Strichmännchen und das zu modellierende System als ein Rechteck. Eine Linie zwischen einem Anwendungsfall und einem Akteur gibt an, dass es zu einer Interaktion kommt. Die «include»-Beziehung zwischen Anwendungsfällen wird durch einen gestrichelten Pfeil und das entsprechende Schlüsselwort dargestellt (siehe [1]).

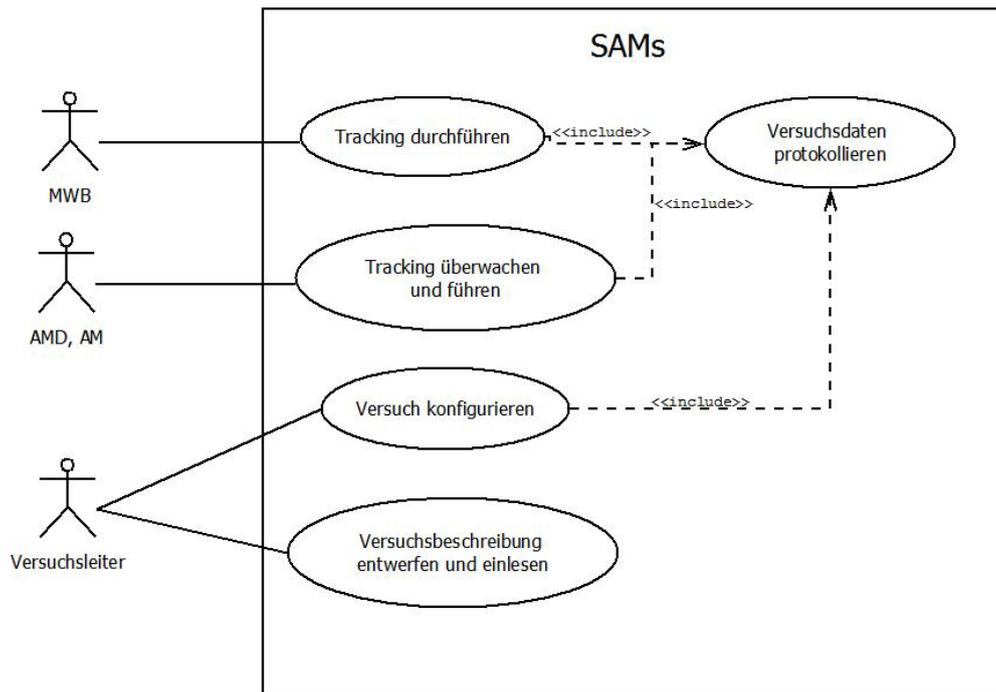


Abbildung 4 - Die Anwendungsfälle von SAMs.

Insgesamt sind fünf Anwendungsfälle in *Abbildung 4* zu sehen, welche in den nächsten Abschnitten im Detail beschrieben werden. Es gibt drei Arten von Akteuren. Zum einen die MWB, welche das Tracking durchführen, die Software-Komponenten AMD und AM, welche das Tracking überwachen und führen sowie den Versuchsleiter, welcher den Versuch entwirft, konfiguriert und steuert.

4.2 TRACKING DURCHFÜHREN

4.2.1 ÜBERBLICK

Der Anwendungsfall *Tracking durchführen* beschreibt die Benutzung von SAMs durch die MWB. Diese führen eine Tracking-Aufgabe aus, bei der es darum geht, ein Tracking-Objekt auf einer Spur entlang zu steuern und ggf. auf Hindernisse und Eingriffe des Operateurs oder der Automatik zu reagieren.

Nachdem der Versuch durch den Versuchsleiter konfiguriert wurde (siehe *Abschnitt 4.4*), werden die verfügbaren Anfangsinstruktionen den MWB zur Auswahl gestellt. Haben diese die ihnen zugewiesenen Instruktionen ausgewählt, gelesen und mit Hilfe eines „Weiter“-Buttons quittiert, beginnt ein Countdown und nach dessen Ablauf startet die Tracking-Aufgabe.

Das Tracking kann kooperativ durchgeführt werden, so dass jeder der MWB einen gewissen Einfluss auf die Steuerung ausüben kann. Hierbei besitzen beide MWB zunächst 50% der Steuergewalt, dieses Verhältnis kann jedoch beeinflusst werden (siehe *Abschnitt 4.3*).

Als Feedback für die aktuelle Geschwindigkeit wird den MWB eine entsprechende Anzeige in Form eines Kreisdiagramms dargeboten. Hierbei nimmt das Diagramm entgegen des Uhrzeigersinns zu, wenn die Geschwindigkeit erhöht wird und nimmt mit dem Uhrzeigersinn ab, wenn die Geschwindigkeit verringert wird. Bei maximaler Geschwindigkeit ist das Diagramm komplett schwarz, bei minimaler Geschwindigkeit komplett grün. Die Geschwindigkeitsanzeige ist hierbei in das Tracking-Objekt integriert.

Eine Strecke gilt als absolviert, wenn das Tracking-Objekt die farbige Zielmarkierung auf der Strecke überfahren oder (falls diese verfehlt wurde) das Ende der Strecke erreicht hat. Nach Abschluss einer Strecke erscheinen

automatisch Endinstruktionen (Versuchshinweise) auf dem Bildschirm, die von den MWB durch das Betätigen eines Buttons bestätigt werden müssen. Diese Hinweise können hierbei von dem Versuchsleiter beim Versuchsentwurf festgelegt werden (siehe Abschnitt 4.5).

4.2.2 EINGABERAUM DER STEUEREINGABEN

Die Steuerung des Tracking-Objekts kann durch Joysticks oder durch die Tastatur vorgenommen werden, wobei letzteres nur für Testzwecke der Entwickler gedacht ist.

STEUERUNG VIA JOYSTICK

Der Eingabedatenraum für die Steuerung via Joystick ist in *Abbildung 5* dargestellt. Hierbei liegen die erwarteten Eingabewerte der horizontalen Achse im Intervall $[-1024, 1023]$ und bei der vertikalen Achse im Intervall $[-1023, 1024]$. Auslenkungen auf der horizontalen Achse des Joysticks führen zu einer horizontalen Bewegung des Fahrzeuges auf dem Bildschirm. Wird der Joystick auf der vertikalen Achse nach vorn bewegt, so wird die Geschwindigkeit des Tracking-Objekts erhöht, zieht man den Joystick zurück, so wird die Geschwindigkeit des Tracking-Objekts reduziert.

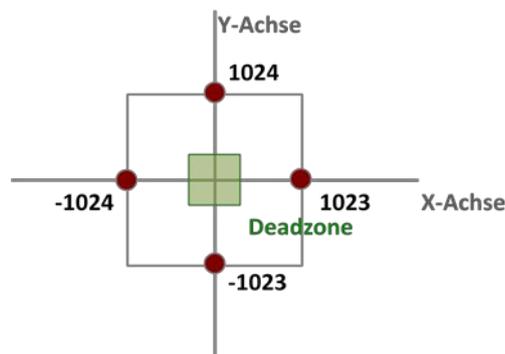


Abbildung 5 - Der Wertebereich eines Joysticks. Wird der Joystick nach links oder vorn ausgelenkt, so ist der Wertebereich $[-1024, 0]$, wird er hingegen nach rechts oder hinten ausgelenkt, so ist der Wertebereich $[0, 1024]$.

Weiterhin existiert eine sogenannte *Deadzone*. Diese definiert den minimalen Ausschlag, welcher überschritten werden muss, damit Steuerinput von ungleich 0 anliegt. Auf diese Weise wird sichergestellt, dass in der Ruhelage des Joysticks immer 0 vorherrscht. Standardmäßig liegt die Deadzone im Bereich $[-30, 30]$.

DAS BEWEGUNGSMODELL

Der *befahrbare Raum* während der Tracking-Simulation wird in *Abbildung 6* dargestellt. Das graue Rechteck bezeichnet die gesamte Strecke, welche während eines Versuchsschrittes absolviert wird, vom Start bis zum Ziel. Der auf dem Bildschirm sichtbare Bereich ist als grünes Rechteck dargestellt. In diesem befindet sich das Tracking-Objekt und kann sich horizontal bis an dessen Ränder bewegen. Die vertikale Position des Tracking-Objekts auf dem Bildschirm ist zwar fest, jedoch ändert sich diese relativ zur Gesamtstrecke. Dies liegt darin begründet, dass während des Trackings der sichtbare Bereich über die Gesamtstrecke geschoben wird, um so den Eindruck einer Fahrt zu erzeugen (vgl. *Abschnitt 4.5*).

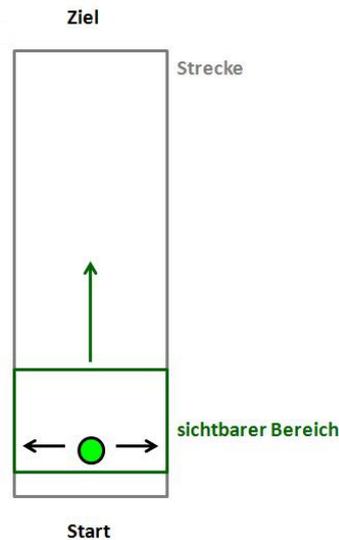


Abbildung 6 - Der befahrbare Bereich von SAMs.

Die Eingaben von den Steuergeräten der MWB bewirken eine Änderung der Position des Tracking-Objekts relativ zur Gesamtstrecke. Im Weiteren wird davon ausgegangen, dass $n \in \mathbb{N}$ MWB an einem Versuch teilnehmen, wobei \mathbb{N} die Menge der natürlichen Zahlen bezeichnet.¹

Die Änderung der horizontalen Position des Tracking-Objekts $xpos^A$ wird wie folgt berechnet:

$$xpos^A = \frac{\sum_{i=1}^n \overline{input}_i^x \cdot alloc_i}{50 \cdot scale_x},$$

wobei \overline{input}_i^x das effektive Input von MWB i darstellt und wie folgt berechnet wird:

$$\overline{input}_i^x = \begin{cases} 0 & , falls (input_i^x > 0) \wedge (denyRight = true) \\ 0 & , falls (input_i^x < 0) \wedge (denyLeft = true) . \\ input_i^x & sonst \end{cases}$$

Dabei bezeichnet $input_i^x$ das direkte Input von MWB i , welches vom Joystick direkt abgelesen wird. Die Variablen $denyRight, denyLeft \in \{true, false\}$ geben an, ob das Lenken nach rechts bzw. links blockiert werden soll. Diese Variablen repräsentieren den Einfluss der harten Eingriffe durch den Operateur bzw. der Automaten (siehe *Abschnitt 4.3.3*).

Weiterhin bezeichnet $alloc_i$ den prozentualen Steuerungsanteil von MWB i (mit $\sum_{i=1}^n alloc_i = 1$) und $scale_x \in \mathbb{R}$ den Dämpfungsfaktor² des horizontalen Steuerinputs darstellen. Die x-Koordinate der neuen Position des Tracking-Objekts $xpos^{neu}$ ergibt sich dann aus der Summe der alten x-Koordinate $xpos^{alt}$ sowie der ermittelten Änderung $xpos^A$:

$$xpos^{neu} = xpos^{alt} + xpos^A.$$

Nahezu die gleiche Formel wird für die vertikale Positionsänderung des Tracking-Objekts benutzt. Hier gibt es eine kleine Besonderheit zu beachten. Die y-Koordinate der Tracking-Objekt-Position ist zwar relativ zum sichtbaren Bereich (also auf dem Bildschirm) konstant, relativ zur Gesamtstrecke verändert sich jedoch diese y-Koordinate. Dies liegt daran, dass die Animation der Strecke durch die Verschiebung des Streckenbildes erreicht wird. Weitere Details werden in *Abschnitt 4.5* dargestellt.

¹ Gegenwärtig nehmen immer 1-2 MWB am Versuch teil, es gilt also $n = 1,2$.

² Dieser Faktor hat die Funktion das Steuerinput zu dämpfen, damit diese nicht zu empfindlich auf die Handbewegungen der MWB reagieren.

Um die Anzahl der Pixel zu berechnen, um die sich das Tracking-Objekt relativ zur Gesamtstrecke vorwärts bewegt ($ypos^{\Delta}$), wird die folgende Formel verwendet:

$$ypos^{\Delta} = \min \left(\frac{\sum_{i=1}^n (1024 - input_i^y) \cdot alloc_i}{50 \cdot scale_y}, ypos_{max}^{\Delta} \right),$$

wobei $ypos_{max}^{\Delta}$ die maximal mögliche Änderung der y-Koordinate in einem konkreten Simulationsschritt darstellt und wie folgt bestimmt wird:

$$ypos_{max}^{\Delta} = v_{max} \cdot v_{set}.$$

Dabei bezeichnet die Konstante $v_{max} = 20.48$ die absolute maximale Geschwindigkeit des Tracking-Objekts (in Pixeln je Simulationsschritt) und $v_{set} \in [0,1]$ die prozentuale maximale Geschwindigkeit des Tracking-Objektes. Erstere ist hierbei konstant für einen Versuch und letztere repräsentiert den Einfluss der harten Eingriffe durch den Operateur bzw. der Automaten (siehe Abschnitt 4.3.3).

Weiterhin bezeichnet $input_i^y$ die vertikale Auslenkung des Steuergerätes von MWB i , $alloc_i$ den prozentualen Steuerungsanteil von MWB i (mit $\sum_{i=1}^n alloc_i = 1$) und $scale_y \in \mathbb{R}$ einen Dämpfungsfaktor des vertikalen Steuerinputs. Die Details zum Streckenaufbau sind Abschnitt 4.5 zu entnehmen.

Aus den obigen Formeln lassen sich zwei Dinge ableiten. Zum ersten bewegt sich das Tracking-Objekt mit der mittleren Geschwindigkeit vorwärts, wenn sich das Steuergerät in der Ruheposition befindet, also keine Eingabe erfolgt und zum zweiten ist kein Rückwärtsfahren möglich, da bei maximaler rückwertiger Auslenkung der Wert $input_i^y = 1024$ erreicht wird (siehe Abbildung 5) und somit $ypos_{T0}^{\Delta} = 0$ gilt.

4.3 TRACKING ÜBERWACHEN UND FÜHREN

4.3.1 ÜBERBLICK

Dieser Anwendungsfall beschreibt die Möglichkeiten den Tracking-Prozess zu überwachen und in diesen einzugreifen. Bei den Akteuren handelt es sich hierbei um die Software-Systeme AMD und AM, welche ebenfalls zum ATEO-System gehören (siehe hierzu Abschnitt 6.2.1). Für die Überwachung des Tracking-Prozesses stellt SAMs diverse Simulationsdaten bereit (siehe Abschnitt 4.3.2) und bietet zudem die Möglichkeit sowohl weiche als auch harte Eingriffe in den Tracking-Prozess zu tätigen (siehe Abschnitt 4.3.3).

Weitere Details zu den anderen beiden Software-Komponenten und zur Vernetzung sind in [6], [7] und [8] zu finden.

4.3.2 ZUGRIFF AUF DIE SIMULATIONS DATEN

Damit die anderen Komponenten den aktuellen Zustand der Simulation erfassen können, stellt SAMs die Simulationsdaten über TCP- und UDP-Sockets bereit. Dies beinhaltet sowohl die Daten zum Simulationszustand als auch die Daten zur Versuchskonfiguration (siehe Tabelle 4-1).

| Name | Bedeutung |
|-----------------------|---|
| inputMwiOne | Steuerungsanteil von MWB 1 |
| obstacleConfig | Konfiguration der statischen Hindernisse (siehe hierzu Abschnitt 4.5) |
| status | Zustand von SAMs: buildTrack, waiting (vor Beginn des Trackings), countdown, waiting+ (nach dem Erreichen des Ziels). |
| stepNr | Nummer des aktuellen Versuchsschrittes |

| | |
|-----------------------------|--|
| track | Dateiname der aktuellen Strecke |
| collision | Art der Kollision |
| coordinateX | x-Koordinate des Tracking-Objekts |
| coordinateY | y-Koordinate des Tracking Objektes (dies entspricht der bereits zurückgelegten Strecke im laufenden Versuchsschritt) |
| dynamicObstacleSpeed | x-Koordinate des dynamischen Hindernisses |
| joystickMwiOne | Steuerinputs des Joysticks von MWB 1 |
| joystickMwiTwo | Steuerinputs des Joysticks von MWB 2 |

Tabelle 4-1 - Von SAMs bereitgestellte Daten.

4.3.3 EINGRIFFE IN DEN TRACKING-PROZESS

Von SAMs werden zwei Arten von Eingriffen unterstützt: weiche und harte Eingriffe. Unter ersteren versteht man das Geben von Hinweisen, welche auditiver oder visueller Natur sein können. Letztere sind Eingriffe in den Tracking-Prozess wie z.B. das Manipulieren des Steuerungsanteils zwischen MWB oder der maximal erreichbaren Geschwindigkeit.

Die möglichen visuellen Hinweise sind in *Abbildung 7* dargestellt. Die Anzeigedauer für einen visuellen Hinweis beträgt hierbei 1.5s. Während dieser Zeit ist es nicht möglich einen weiteren visuellen Hinweis zu geben.

Bevor ein auditiver Hinweis gegeben werden kann, muss zunächst gewählt werden, ob nur MWB1, nur MWB2 oder beide MWB den Hinweis erhalten sollen. Die Adressierung des auditiven Hinweises wird hierbei über die entsprechende Ansteuerung der beiden Stereo-Kanäle der Soundausgabe und unter Verwendung von Kopfhörern realisiert. *Tabelle 4-2* listet die möglichen auditiven Hinweise auf.

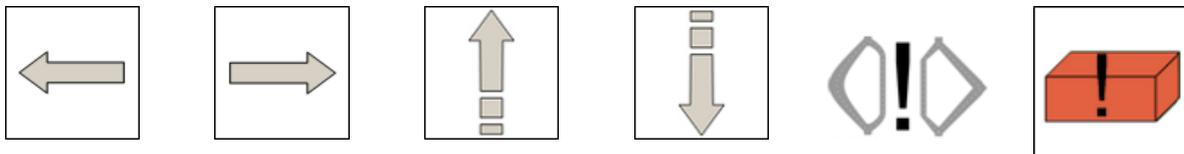


Abbildung 7 - Mögliche visuelle Hinweise (v.l.n.r.): "Nach links fahren", "Nach rechts fahren", "Geschwindigkeit erhöhen", "Geschwindigkeit verringern", „Achtung Gabelung“ und „Achtung Hinderniss“.

| Sound | Hinweis |
|----------------------------------|------------------------|
| fuehrung ueberlassen.mp3 | Führung überlassen! |
| fuehrung uebernehmen.mp3 | Führung übernehmen! |
| kurven abkuerzen.mp3 | Kurven abkürzen! |
| kurven genauer fahren.mp3 | Kurven genauer fahren! |
| kurven schneiden.mp3 | Kurven schneiden! |
| langsamer fahren.mp3 | Langsamer fahren! |
| links fahren.mp3 | Linsk fahren! |
| rechts fahren.mp3 | Rechts fahren! |
| schneller fahren | Schneller fahren! |
| weiter so.mp3 | Weiter so! |

Tabelle 4-2 - Mögliche auditive Hinweise.

Insgesamt stehen drei Arten von harten Eingriffen zur Verfügung. Es kann die maximal erreichbare Geschwindigkeit des Tracking-Objekts in 1%-Schritten manipuliert werden, wobei Werte von 0% bis 100% möglich sind. Zudem kann das Lenken nach links oder nach rechts blockiert werden, so dass das Lenken in die blockierte Richtung effektiv auf die Position des Tracking-Objekts ist. Weiterhin ist es möglich, die Steuergewaltauftei-

lung in 5%-Schritten zwischen den MWB zu verändern, so dass das Steuerinput des einen MWB stärker gewichtet wird als das Steuerinput des anderen. Hierbei ist auch möglich, einem MWB 100% und dem anderen 0% der Steuergewalt zuzuweisen.

Die gewünschten Eingriffe für auditive und visuelle Hinweise werden über *Codes* von SAMs bereitgestellt, welche in *Tabelle 4-3* dargestellt sind. Ein Logging geschieht separat in verschiedenen Variablen (siehe *Abschnitt 4.6*).

| Code | Bedeutung |
|------|--|
| 1 | Linker Stereokanal gewählt |
| 2 | Rechter Stereokanal gewählt |
| 3 | Beide Stereokanäle gewählt |
| 21 | Auditiver Hinweis „langsamer fahren“ |
| 22 | Auditiver Hinweis „schneller fahren“ |
| 31 | Auditiver Hinweis „links fahren“ |
| 32 | Auditiver Hinweis „rechts fahren“ |
| 41 | Auditiver Hinweis „Kurven abkürzen“ |
| 42 | Auditiver Hinweis „Kurven genauer fahren“ |
| 51 | Auditiver Hinweis „Führung überlassen“ |
| 52 | Auditiver Hinweis „Führung übernehmen“ |
| 61 | Auditiver Hinweis „Weiter so“ |
| 69 | Visueller Hinweis „Hindernis“ |
| 79 | Visueller Hinweis „Gabelung“ |
| 81 | Visueller Hinweis „Geschwindigkeit verringern“ |
| 82 | Visueller Hinweis „Geschwindigkeit erhöhen“ |
| 91 | Visueller Hinweis „Links halten“ |
| 92 | Visueller Hinweis „Rechts halten“ |

Tabelle 4-3 - Codes für das Auslösen von auditiven und visuellen Hinweisen.

4.3.4 FEHLBEDIENUNG

Bei den weichen Eingriffen kann es zu Fehlbedienungen kommen. So ist es nicht möglich einen auditiven bzw. visuellen Hinweis zu geben solange noch ein anderer auditiver bzw. visueller Hinweis aktiv ist.

4.4 VERSUCH KONFIGURIEREN

4.4.1 ÜBERBLICK

Bevor der Versuch beginnt, besitzt der Versuchsleiter die Möglichkeit versuchsrelevante Daten zu konfigurieren. Hierfür steht ein GUI-Dialog zur Verfügung, welcher genauer in *Abschnitt 4.4.2* beschrieben wird. Die Versuchskonfiguration wird ausschließlich vom Versuchsleiter durchgeführt. Die Versuchskonfiguration wird hierbei durch das Betätigen des Buttons „Versuch starten“ abgeschlossen und das Tracking kann nun durchgeführt werden (siehe hierzu *Abschnitt 4.2*).

4.4.2 FUNKTIONALE BESCHREIBUNG

Dem Versuchsleiter steht zu Beginn des Versuches ein Menü zur Verfügung, in dem verschiedene Einstellungen getätigt werden können. *Abbildung 8* zeigt den Aufbau des Konfigurationsmenüs mit dem die folgenden Einstellungen möglich sind:

1. **Versuchsbedingung (Element 1)**
Den Versuchsbedingungen werden eindeutige Nummern zugewiesen.
2. **Eingabe der Teamnummer (Element 2)**
Den zusammen fahrenden MWB wird eine eindeutige Nummer zugewiesen.
3. **Eingabe Versuchspersonennummern von MWB 1 und 2 (Element 3 und 4)**
Jeder Versuchsperson wird eine eindeutige Zahl zugeordnet.
4. **Eingabe der Nummer für die Assistenz (Element 5)**
Der Assistenz wird ebenfalls eine eindeutige Nummer zugewiesen.
5. **Eingabe der Art der Assistenz (Element 6)**
Die Assistenz kann entweder durch einen Operateur oder aber durch Automaten ausgeführt werden.
6. **Eingabe des Geschlechts (Element 7)**
Ein Versuchsteam kann entweder männlich oder weiblich sein.
7. **Netzwerk (Element 8)**
Die Netzwerkunterstützung kann ein (grün) oder aus (rot) geschaltet werden.
8. **Versuch starten (Element 9)**
Mit diesem Button kann der Versuch gestartet werden.

The image shows a configuration menu titled "SAM Konfiguration". It contains several rows of controls, each with a numbered callout box to its right:

- Row 1: "- 1 +" followed by "Versuchsbedingung" and a box with "1".
- Row 2: "- 1 +" followed by "Nummer Team" and a box with "2".
- Row 3: "- 1 +" followed by "Nummer MWB1" and a box with "3".
- Row 4: "- 1 +" followed by "Nummer MWB2" and a box with "4".
- Row 5: "- 1 +" followed by "Nummer Assistenz" and a box with "5".
- Row 6: "Operateur" (selected), "Automatik", "Assistenz" and a box with "6".
- Row 7: "Männlich" (selected), "Weiblich", "Geschlecht" and a box with "7".
- Row 8: "Netzwerk" (highlighted in red), a box with "8".
- Row 9: "Versuch starten" and a box with "9".

Abbildung 8 - Konfigurationsmenü für den Versuchsleiter.

4.4.3 BESCHREIBUNG DER DATEN

Die per Dialog konfigurierbaren Daten werden einmalig in einer separaten Logdatei erfasst und sind in *Tabelle 4-4* dargestellt.

| Datum | Bedeutung |
|--------------------------|--|
| Versuchsbedingung | Jeder Versuchsbedingung wird eine eindeutige Nummer zugewiesen, um diese in der folgenden Analyse unterscheiden zu können. |
| Nummer Team | Jedem Paar von MWB, welche zusammen an einem Versuch teilnehmen wird eine eindeutige Nummer zugewiesen. Somit können weitere Testbedingungen erfasst werden und in der anschließenden Analyse unterschieden werden (analog zur Versuchsbedingung). |
| Nummer MWB1, MWB2 | Jeder MWB bekommt eine eindeutige Nummer zugewiesen, welche die MWB in der anschließenden Analyse unterscheidbar macht. |
| Nummer Assistenz | Jeder Assistenz (Automatik wie Operateur) wird ebenfalls eine eindeutige Nummer zugewiesen. Dies ermöglicht eine Unterscheidung bei der Analyse der Versuchsergebnisse. |
| Assistenz | Die Art der Assistenz. Um in der Analyse unterscheiden zu können, ob eine Automatik oder aber ein Operateur assistiert hat. |
| Geschlecht | Das Geschlecht des aktuellen Versuchsteams. |

Tabelle 4-4 - Die konfigurierbaren Versuchsdaten.

4.5 VERSUCHSBESCHREIBUNG ENTWERFEN UND EINLESEN

4.5.1 ÜBERBLICK

SAMs bietet die Möglichkeit den Versuch mit Hilfe von externen Textdateien zu konfigurieren. Ein Versuch besteht aus mehreren Versuchsschritten. In jedem Versuchsschritt haben die MWB eine Tracking-Aufgabe zu absolvieren. Hierbei ist es sowohl möglich die Strecke (siehe *Abschnitt 4.5.2*) als auch einige Parameter zur Tracking-Aufgabe der jeweiligen Versuchsschritte (siehe *Abschnitt 4.5.3*) zu konfigurieren.

4.5.2 STRECKENBESCHREIBUNG

Eine Strecke besteht aus einer Aneinanderreihung von vorgegeben Spielbrettern, welche mit Hilfe eines Bildbearbeitungsprogrammes zu einer Bilddatei zusammengesetzt wird.

Abbildung 9 zeigt einige Beispiele der verfügbaren Spielbretter. Das Ende einer Strecke wird hierbei immer durch ein spezielles Ziel-Spielbrett festgelegt, welches in *Abbildung 2* zu sehen ist. Weitere Informationen zu den Spielbrettern, wie z.B. welche Spielbretter zur Verfügung stehen und welche Farben für die Erstellung von Spielbrettern verwendet werden sollen, sind in [9] zu finden.

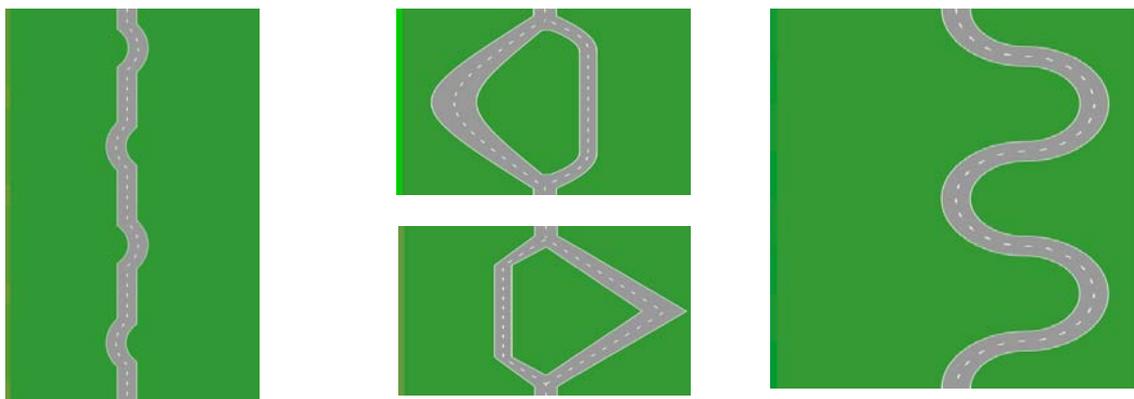


Abbildung 9 – Beispiele für die verschiedenen Spielbretter.

Zur Unterscheidung der einzelnen Abschnitte auf einen Spielbrett wurden am linken Rand Informationen in Form von Rot-Werten codiert. Diese Werte geben Auskunft darüber, wie die Strecke beschaffen ist, so kann z.B. ermittelt werden, ob es sich um einen kurvigen Teil oder eine gerade Strecke handelt. Eine Übersicht über die verwendeten Rot-Werte ist ebenfalls [9] zu entnehmen.

HINDERNISSE

Auf der Strecke können zudem sowohl statische als auch dynamische Hindernisse platziert werden. Die statischen Hindernisse sind in einem Slalom aus zwei Hindernissen angeordnet und unterscheiden sich in dem Grad der Straßenabdeckung, welcher 25% oder 50% betragen kann. Zudem kann ein solcher Slalom mit einem links platzierten oder einem rechts platziertes Hindernis beginnen. Das darauffolgende Hindernis befindet sich dann jeweils auf der anderen Seite der Strecke.

Das dynamische Hindernis bewegt sich stets von links nach rechts über das Bild, wobei die Geschwindigkeit des Hindernisses auf das Fahrverhalten der MWB so angepasst ist, dass ohne eine Änderung des Fahrverhaltens, also ohne ein Erhöhen oder Verringern der aktuellen Geschwindigkeit, eine Kollision unausweichlich ist. Auf Spielbrettern mit einem dynamischen Hindernis wird deswegen die Geschwindigkeit der MWB gemessen und für das folgende dynamische Hindernis genutzt, wobei dieses dynamische Hindernis noch im gleichen oder

folgenden Versuchsschritt auftreten kann. Beim ersten dynamischen Hindernis eines Versuches wird die mittlere Geschwindigkeit vorgegeben, die von den MWBs erreicht wird, wenn auf der vertikalen Achse des Joysticks kein Input gegeben wird, d.h. weder gebremst noch beschleunigt wurde.

Die Position der statischen und dynamischen Hindernisse wird in externen Textdateien definiert. Jede Zeile stellt hierbei einen Eintrag da, wobei die Einträge die folgende Gestalt haben:

HindernisCode; Offset

wobei

- *HindernisCode*: beschreibt die Art des Hindernisses. Die möglichen Werte werden in *Tabelle 4-5* beschrieben.
- *Offset*: Anzahl der Pixel, die zwischen dem oberen Ende der Strecke und dem oberen Ende des Spielbretts liegen, auf dem das Hindernis erscheinen soll.

Tabelle 4-5 listet die verfügbaren Hindernisse mit den dazugehörigen Codes auf.

| Wert | Bedeutung |
|-------------|--|
| 1100 | Erstellung eines statischen Slaloms mit 25% Fahrbahnabdeckung und dem ersten Hindernis rechts. |
| 1110 | Erstellung eines statischen Slaloms mit 25% Fahrbahnabdeckung und dem ersten Hindernis links. |
| 1101 | Erstellung eines statischen Slaloms mit 50% Fahrbahnabdeckung und dem ersten Hindernis rechts. |
| 1111 | Erstellung eines statischen Slaloms mit 50% Fahrbahnabdeckung und dem ersten Hindernis links. |
| 111 | Erstellung eines dynamischen Hindernisses. |

Tabelle 4-5 - Codes für die Konfiguration von Hindernissen.

4.5.3 VERSUCHSBESCHREIBUNG

Nachdem die Strecken erstellt wurden, muss noch festgelegt werden, in welchem Versuchsschritt die jeweilige Strecke abgefahren werden soll. Dies passiert in der Text-Datei *steps.txt*. Dieser Name ist fest vorgegeben und muss eingehalten werden. *Abbildung 10* zeigt wie eine solche Datei aussehen kann.

```
1; 1; lernstrecke; hindernisKonfiguration_0.txt;automatik1.aaf
2; 1; testabschnitt; hindernisKonfiguration_0.txt
3; 2; lernstrecke; hindernisKonfiguration_0.txt;automatik2.aaf
4; 2; testabschnitt; hindernisKonfiguration_0.txt
5; 12; hauptabschnitt_ohne_manipulationen; hindernisKonfiguration_0.txt
6; 12; hauptabschnitt_ohne_manipulationen; hindernisKonfiguration_0.txt
7; 12; hauptabschnitt1; hindernisKonfiguration_1.txt
8; 12; hauptabschnitt2; hindernisKonfiguration_2.txt
9; 12; hauptabschnitt_lang; hindernisKonfiguration_7.txt
10; 12; hauptabschnitt_lang; hindernisKonfiguration_7.txt
11; 12; hauptabschnitt_lang; hindernisKonfiguration_7.txt
```

Abbildung 10 – Ein Beispiel für die Konfigurationsdatei *steps.txt*.

Ein Eintrag in die Datei *steps.txt* hat dabei die folgende Struktur:

StepNr ; SteuerInput ; Strecke; Hindernisse; Automatik

wobei folgendes gilt:

- *StepNr*: die Nummer des Versuchsschritts.
- *SteuerInput*: Verteilung des Steuer-Inputs. Folgende Werte sind hier möglich:
 - 1: Der erste MWB erhält 100% der Steuergewalt.
 - 2: Der zweite MWB erhält 100% der Steuergewalt.
 - 12: Beide MWB erhalten 50% der Steuergewalt.
- *Strecke*: der Name der Bilddatei (ohne Endung), welche in diesem Versuchsschritt geladen werden soll
- *Hindernisse*: der Name derjenigen Textdatei (ohne Endung), welche die Hindernisse der Strecke konfiguriert (siehe *Abschnitt Hindernisse*).
- *Automatik*: der Name der xml-Datei (Endung aaf), welche die Automatik für die Strecke konfiguriert. Die Angabe einer Automatik ist optional, wird keine angegeben, so wird die leere Automatik verwendet, falls Automatik als Assistenz gewählt wurde im Konfigurationsmenü. Die Automatik reicht alle Werte unverändert durch.

4.6 VERSUCHSDATEN PROTOKOLLIEREN

4.6.1 ÜBERBLICK

Die bei einem Versuch aufgezeichneten Daten dienen zur Verhaltensauswertung der Versuchspersonen und des Operators bzw. der Automaten. Der ATEO-Versuch ist in Schritte untergliedert. Je Versuchsschritt wird eine Datei im CSV-Format erzeugt. Diese Datei wird als Log-File bezeichnet. Die Versuchsaufzeichnung beginnt mit dem Starten der Fahrt und wird periodisch wiederholt.

4.6.2 FUNKTIONALE BESCHREIBUNG

Für jeden Versuchsschritt wird eine eigene Protokolldatei angelegt. Startet man eine Fahrt neu, so wird eine weitere Protokolldatei angelegt. Dies spiegelt sich auch in der Benennung der Protokolldateien wider. Wie diese genau aussieht wird weiter unten in diesem Abschnitt beschrieben.

Am Anfang eines Experimentes wird die vom Versuchsleiter im Konfigurationsmenü eingestellte Konfiguration in eine XML-Datei geschrieben. Zu Beginn eines jeden Versuchsschritts wird dann in diese XML-Datei die Konfiguration des Versuchsschrittes geschrieben. Anschließend werden die Variablen, die während der Fahrübung erzeugt werden, in einer zusätzlichen Protokolldatei vermerkt.

Es gibt zwei Daten-Quellen für die Protokolldatei: Daten von der Steuerung durch die MWB, Daten von den Hinweisen des Operators.

BENENNUNG DER PROTOKOLLDATEIEN

Der Name der Protokolldateien setzt sich wie folgt zusammen

Datum, '_ ', Uhrzeit, '_ ', Stepnr, '.csv'

wobei

- *'_ '* und *'.csv'* Zeichenketten darstellen, welche direkt in den Dateinamen eingehen. *'.csv'* ist das typische Suffix für CSV-Dateien und der Unterstrich wird als Separator einzelner Blöcke im Dateinamen benutzt.
- Der erste Block besteht aus dem Datum zum Zeitpunkt der Erstellung (Jahr/Monat/Tag).

- Der zweite Block besteht aus der Uhrzeit zum Zeitpunkt der Erstellung (Stunde/Minute/Sekunde).
- *StepNr* ist die Nummer des aktuellen Versuchsschritts.

Der Name der Konfigurationsdatei setzt sich wie folgt zusammen

$$Datum, '_ ', Uhrzeit, '_ ', 'config', '.xml'$$

wobei sich der Dateiname von der Protokolldatei nur durch den Dateitypen (XML-Datei) und dem Anhängen der Zeichenkette 'config' anstelle der *StepNr* unterscheidet.

4.6.3 BESCHREIBUNG DER DATEN

Die Daten werden in zwei Dateien aufgeteilt. In einer XML-Datei werden die über ein Experiment konstanten Konfigurationsdateien gespeichert. In einer zweiten Datei werden die in einem Versuchsschritt anfallenden Daten geloggt. Diese lassen sich in die Kategorien Systemzustände, Ereignisse und Leistungsmaße unterteilen.

KONFIGURATIONSDATEI

Die Daten dieser Konfigurationsdatei dienen der Beschreibung des Versuchs im Großen.

| Variable | Datentyp | Definitionsbereich | Beschreibung |
|------------------|----------|----------------------|---|
| experimentNr | Integer | beliebig | Nummer der Versuchsgruppe. |
| teamNr | Integer | beliebig | Nummer des Teams. |
| mwi1 | Integer | beliebig | Nummer des Mikroweltbewohners 1. |
| mwi2 | Integer | beliebig | Nummer des Mikroweltbewohners 2. |
| assistanceNr | Integer | beliebig | Nummer des Operateurs |
| kindOfAssistance | String | Operateur, Automatik | Art der genutzten Assistenz |
| gender | String | Male, female | Geschlecht der Versuchsgruppe |
| Nr | Integer | beliebig | Nummer des aktuellen Versuchsschritts. |
| mwMWIOneOnOP | Integer | [0..150] | Mental Workload des MWB1, wie er vor dem Step gemessen wurde |
| mwMWITwoOnOP | Integer | [0..150] | Mental Workload des MWB2, wie er vor dem Step gemessen wurde |
| trackname | String | beliebig | Name der Strecke, welche für den Versuchsschritt gefahren werden soll. |
| inputMWI1 | Real | [0..1] | Vor dem Versuchsschritt definierte Inputverteilung des MWB1 (entspricht alloc;). Daraus kann sich die Inputverteilung von MWB2 abgeleitet werden. |

PROTOKOLLDATEI

Die Daten der Protokolldatei lassen sich in die Kategorien Systemzustände, Ereignisse und Leistungsmaße einteilen. Diese werden im Folgenden im Detail beschrieben.

SYSTEMZUSTÄNDE

Die Daten dieser Kategorie beschreiben Systemzustände während des Versuchs, wobei hier noch einmal zwischen Daten bezüglich der Strecke und den Daten, die sich auf das Fahrzeug beziehen unterschieden wird.

| Variable | Datentyp | Definitionsbereich | Beschreibung |
|------------------------------------|----------|--------------------|--|
| ColorCode | Real | [0..255] | Rotfarbwert der aktuellen Streckenposition bei besonderen Streckenabschnitten zur genaueren Positionsbestimmung (RGB-Wert) |
| FirstStaticObstacleXCoordinate | Real | beliebig | x-Koordinate des ersten statischen Hindernisses |
| FirstStaticObstacleYCoordinate | Real | beliebig | y-Koordinate des ersten statischen Hindernisses |
| SecondStaticObstacleXCoordinate | Real | beliebig | X-Koordinate des zweiten statischen Hindernisses |
| SecondStaticObstacleYCoordinate | Real | beliebig | Y-Koordinate des zweiten statischen Hindernisses |
| DynamicObstacleXCoordinate | Real | beliebig | X-Koordinate des dynamischen Hindernisses |
| DynamicObstacleYCoordinate | Real | beliebig | Y-Koordinate des dynamischen Hindernisses |
| DynamicObstacleDistanceTick | Real | [0..20,48] | Geschwindigkeit des dynamischen Hindernisses |
| TrackEditMWIVisibleStaticObstacle | Integer | {0, 1} | 0 := statisches Hindernis ist nicht sichtbar für die MWBs 1 := statisches Hindernis ist sichtbar für die MWBs |
| TrackEditMWIVisibleDynamicObstacle | Integer | {0, 1} | 0 := dynamisches Hindernis ist nicht sichtbar für die MWBs 1 := dynamisches Hindernis ist sichtbar für die MWBs |
| TrackEditMWIVisibleFork | Integer | {0, 1} | 0 := Gabelung ist nicht sichtbar für die MWBs 1 := Gabelung ist sichtbar für die MWBs |
| TrackEditOAVisibleStaticObstacle | Integer | {0, 1} | 0 := statisches Hindernis ist nicht sichtbar für den Operateur 1 := statisches Hindernis ist sichtbar für den Operateur |
| TrackEditOAVisibleDynamicObstacle | Integer | {0, 1} | 0 := dynamisches Hindernis ist nicht sichtbar für den Operateur 1 := dynamisches Hindernis ist sichtbar für den Operateur |
| TrackEditOAVisibleFork | Integer | {0, 1} | 0 := Gabelung ist nicht sichtbar für den Operateur 1 := Gabelung ist sichtbar für den Operateur |

Weiterhin fallen Daten an, welche sich auf das **Fahrzeug** beziehen. Diese sind in den folgenden beiden Tabellen beschrieben.

| Variable | Datentyp | Definitionsbereich | Beschreibung |
|----------------------|----------|--------------------|---|
| DistanceTotal | Real | beliebig | aufsummierte zurückgelegte Strecke in vertikaler Richtung des Fahrobjektes (in Pixel) |
| DistanceTick | Real | beliebig | Schrittweite in vertikaler Richtung des Fahrobjektes (in Pixel) |
| XCoordinateOnSection | Real | beliebig | horizontale Position des Fahrobjektes auf dem Streckenbild (in Pixel) |
| YCoordinateOnSection | Real | beliebig | vertikale Position des Fahrobjektes auf dem Streckenbild (in Pixel) |

EREIGNISSE

Die Daten dieser Kategorie entstehen durch auftretende Ereignisse während des Versuchs. Quellen dieser Ereignisse sind sowohl die MWB als auch der Operateur.

Die Ereignisse der MWB sind ihr Lenkverhalten, welche Daten im genauen mitgeschrieben werden, beschreibt die folgende Tabelle:

| Variable | Datentyp | Definitionsbereich | Beschreibung |
|---------------|----------|--------------------|---|
| LeftRightMWI1 | Integer | [-1024,1023] | horizontale Ausrichtung des Joysticks von Mikroweltbewohner 1 |
| LeftRightMWI2 | Integer | [-1024,1023] | horizontale Ausrichtung des Joysticks von Mikroweltbewohner 2 |
| UpDownMWI1 | Integer | [-1024,1023] | vertikale Ausrichtung des Joysticks von Mikroweltbewohner 1 |
| UpDownMWI2 | Integer | [-1024,1023] | vertikale Ausrichtung des Joysticks von Mikroweltbewohner 2 |

Die Ereignisse des Operateurs hingegen resultieren aus den getätigten **Eingriffen** desselbigen.

| Variable | Datentyp | Definitionsbereich | Beschreibung |
|--------------------|----------|--------------------|--|
| AudHintAllocation | Integer | {0, 1, 2 } | 1 := ausgewählte Versuchsperson 1 2 := ausgewählte Versuchsperson 2 3 := ausgewählte Versuchsperson 1+2 |
| AudHintSpeed | Integer | {0, 1, 2 } | 0: kein auditiver Hinweis abgespielt 1: auditiver Hinweis „langsamer fahren“ wurde abgespielt 2: auditiver Hinweis „schneller fahren“ wurde abgespielt |
| AudHintDirection | Integer | {0, 1, 2 } | 0: kein auditiver Hinweis abgespielt 1: auditiver Hinweis „links fahren“ wurde abgespielt 2: auditiver Hinweis „rechts fahren“ wurde abgespielt |
| AudHintAccuracy | Integer | {0, 1, 2 } | 0: kein auditiver Hinweis abgespielt 1: auditiver Hinweis „Kurven abkürzen“ wurde abgespielt 2: auditiver Hinweis „Kurven genauer fahren“ wurde abgespielt |
| AudHintPower | Integer | {0, 1, 2 } | 0: kein auditiver Hinweis abgespielt 1: auditiver Hinweis „Führung überlassen“ wurde abgespielt 2: auditiver Hinweis „Führung übernehmen“ wurde abgespielt |
| AudHintPraise | Integer | {0, 1} | 0: kein auditiver Hinweis abgespielt 1: auditiver Hinweis „Weiter so“ wurde abgespielt |
| DirectSetDirection | Integer | {0, 1, 2, 3} | 0: die Joystickausrückung der MWB wurde nicht vom Operateur in einer Richtung eingeschränkt 1: die Joystickausrückung der MWB wurde vom Operateur in nach links eingeschränkt 2: die Joystickausrückung der MWB wurde vom Operateur in nach rechts eingeschränkt |

| | | | |
|---------------------|---------|------------|---|
| | | | 3: die Beschränkung der Joystickauslenkung wurde vom Operateur wieder zurückgesetzt, so dass die MWB nicht mehr eingeschränkt sind |
| DirectSetPowerMWI1 | Real | [0..1] | Inputverteilung von MWB 2, welcher durch den Operateur angepasst werden kann |
| DirectSetPowerMWI2 | Real | [0..1] | Inputverteilung von MWB 2, welcher durch den Operateur angepasst werden kann |
| DirectSetPowerReset | Integer | {0, 1} | 0: Die Inputverteilung wurde nicht vom Operateur zurückgesetzt 1: Die Inputverteilung der MWB wurde vom Operateur auf 50% für beide MWB zurückgesetzt. |
| DirectSetSpeed | Real | [0..1] | Geschwindigkeit des Trackingobjektes, welche vom Operateur angepasst werden kann |
| DirectSetSpeedReset | Integer | {0, 1} | 0: Die Geschwindigkeit des Trackingobjektes wurde vom Operateurs nicht zurücksetzt 1: Die Geschwindigkeit des Trackingobjektes wurde von Operateur auf 100% zurückgesetzt. |
| VisHintObstacle | Integer | {0, 9} | 0: kein visueller Hinweis wurde angezeigt 9: visueller Hinweis „Hindernis“ wurde angezeigt |
| VisHintFork | Integer | {0, 9} | 0: kein visueller Hinweis wurde angezeigt 9: visueller Hinweis „Gabelung“ wurde angezeigt |
| VisHintSpeed | Integer | {0, 1, 2 } | 0: kein visueller Hinweis wurde angezeigt 1: visueller Hinweis „langsamer fahren“ wurde angezeigt 2: visueller Hinweis „schneller fahren“ wurde angezeigt |
| VisHintDirection | Integer | {0, 1, 2 } | 0: kein visueller Hinweis wurde angezeigt 1: visueller Hinweis „links fahren“ wurde angezeigt 2: visueller Hinweis „rechts fahren“ wurde angezeigt |

Die Blind-Clicks des Operateurs und sein Situationsbewusstsein werden in den folgenden zwei Variablen erfasst:

| Variable | Datentyp | Definitionsbereich | Beschreibung |
|--------------------|----------|--|---|
| BlindClick | Point | Beliebig im Rahmen der Auflösung des AMD (2560x1920) | Die Punkte, auf die der Operateur klickt und die keine Aktion auslösen bzw. ein Fehlverhalten bedeuten, werden in dieser Spalte aufgezeichnet. |
| SituationAwareness | Integer | {0,1,2,3,11,22} | 0: die Messung des Situationsbewusstseins ist nicht aktiv 1: die Messung des Situationsbewusstseins beginnt mit dem Aktivieren des linken Buttons 2: der Operateur hat rechtzeitig auf den linken Button geklickt, der rechte Button wird aktiv 3: die Messung wurde vom Operateur im vorgegebenen Zeitlimit erfolgreich mit dem Drücken des rechten Buttons beendet 11: der Operateur hat nicht rechtzeitig auf den linken Button geklickt. Die Messung ist damit beendet. 22: Der Operateur hat nicht rechtzeitig auf den rechten Button geklickt, Die Messung ist damit beendet |

LEISTUNGSMASSE

Die Daten der letzten Kategorie dienen als Grundlage für die Bestimmung von Leistungsdaten und sind daher auch so betitelt. Die folgende Tabelle erläutert diese Daten im Detail:

| Variable | Datentyp | Definitionsbereich | Beschreibung |
|-------------------|----------|--------------------|--|
| TimeTotal | Integer | > 0 | Zeit relativ zum Beginn des Versuchs (in ms) |
| TimeDelta | Integer | > 0 | Differenz der aktuellen zur vorhergehenden Zeitmessung (in ms) |
| NoSensorsOffTrack | Integer | [0..8] | Anzahl der Sensoren, die sich auf dem Grün befinden. |
| CollisionAtSensor | Integer | [1..8] | gibt an, welcher Sensor 1-8 eine Kollision gemeldet hat, wobei mehrere Nennungen in einer Zeile möglich sind |

5 NICHTFUNKTIONALE ANFORDERUNGEN & QUALITÄTSZIELE

5.1 ÜBERSICHT

| Produktqualität | sehr gut | gut | normal | nicht relevant |
|------------------------|----------|-----|--------|----------------|
| Funktionalität | | | | |
| Angemessenheit | x | x | | |
| Richtigkeit | x | | | |
| Interoperabilität | | | x | |
| Ordnungsmäßigkeit | | | | x |
| Sicherheit | | | | x |
| Zuverlässigkeit | | | | |
| Reife | | x | | |
| Fehlertoleranz | | | x | |
| Wiederherstellbarkeit | | | x | |
| Benutzbarkeit | | | | |
| Verständlichkeit | | x | | |
| Erlernbarkeit | x | | | |
| Bedienbarkeit | x | | | |
| Effizienz | | | | |
| Zeitverhalten | x | | | |
| Verbrauchsverhalten | x | | | |
| Änderbarkeit | | | | |
| Analysierbarkeit | x | | | |
| Modifizierbarkeit | x | | | |
| Stabilität | x | | | |
| Prüfbarkeit | | | x | |
| Übertragbarkeit | | | | |
| Anpassbarkeit | | x | | |
| Installierbarkeit | | | | x |
| Konformität | | x | | |
| Austauschbarkeit | | x | | |

5-1 - Qualitätsziele von SAMs in der Übersicht.

5.2 ZUVERLÄSSIGKEIT / EFFIZIENZ

Die mittels SAMs im Rahmen der Forschung gewonnenen Ergebnisse wären wertlos, wenn das System die Anforderungen der Zuverlässigkeit und Effizienz vernachlässigen würde. So ist es entscheidend, dass die in der Protokolldatei zu erfassenden Daten kontinuierlich und innerhalb des Intervalls [30,49] ms erhoben werden. Wird die vorgegebene Zeitspanne zwischen dem Abfragen von Daten lastbedingt erheblich überschritten oder ist diese sehr unregelmäßig, so kann es zu Ungenauigkeiten und Verzerrungen in der Protokolldatei kommen und zudem kann dies zu Verzögerungen während der Tracking-Simulation kommen, welche für die MWB erkennbar sind und somit einen ungewünschten Einfluss auf die MWB ausüben. Dies kann zu unbrauchbaren Versuchsdaten führen. Ebenso führen unplanmäßige Programmabbrüche zu unbrauchbaren Versuchsdaten und sind damit nach Möglichkeit zu vermeiden.

5.3 BEDIENBARKEIT

Da es sich bei den Probanden (Mikroweltbewohnern) nicht unbedingt um PC-erfahrene Personen handelt, muss die Bedienung der Software, insbesondere die Instruktionsmasken und die Steuerung der Navigationshinweise, einfach zu handhaben sein. Auch wenn die Tracking-Aufgabe keine realistische Fahrsimulation darstellt, sollte die Fahrobjektsteuerung nachvollziehbar und leicht erlernbar sein.

Weiterhin der Versuchsleiter bei der Konfiguration und Steuerung des Versuchs von der Software unterstützt werden.

5.4 WARTBARKEIT

Das Projekt ATEO entwickelt sich ständig weiter und somit ist auch das System ATEO ständigen Änderungen ausgesetzt. Deswegen ist die Erweiterbarkeit des Systems eine ausschlaggebende Anforderung. Der Entwurf von allgemeinen Interfaces und Bibliotheken mit wieder verwendbaren Funktionen soll den Entwicklungsprozess vereinfachen.

5.5 VARIABILITÄT

Das Tracking ist momentan fix, d.h. die Position, Farbe, Größe, ... ist fest im Projekt verankert. Wenn man diese festen Vorgaben ebenfalls konfigurierbar macht, eröffnen sich weitere Testmöglichkeiten und Einsatzgebiete für ATEO.

6 TECHNISCHE SYSTEMUMGEBUNG

6.1 REFERENZSYSTEM

6.1.1 HARDWARE

Das ATEO-System wird momentan erfolgreich auf dem folgenden System eingesetzt:

- Intel Core2 Quad Q6600 @ 2.40 Ghz (4x 2.40 Ghz)
- 3GB DDR2 1066Mhz (3x1GB)
- Dell RV770 (ATI Radeon HD 4850)
- Mainboard: Dell 0G254H mit Dell X48 Processor (FSB 4x 266Mhz DDR2/3 2x 532Mhz)
- 640 GB Western Digital WDC WD6400AAKS-75A7B (SATA300)

Die Peripherie umfasst wie bereits in *Abbildung 12* angegeben:

- 2 Bildschirme (19" und 30")
- Maus zur Versuchssteuerung, Operateursarbeitsplatz
- Maus für das Versuchspersonenpaar (ohne mittlere Maustaste)
- Lautsprecher zur Ausgabe von Anweisungen
- 2 Kopfhörer für das Versuchspersonenpaar
- 2 Joysticks

6.1.2 SOFTWARE

SAMs wurde in Smalltalk entwickelt und wird daher in der Entwicklungs- und Laufzeitumgebung Squeak ausgeführt. Diese wird von einer virtuellen Maschine ausgeführt, welche für verschiedenste Plattformen entwickelt wurde³. Im Referenzsystem kommt ein Windows Vista 32 Bit zum Einsatz.

6.2 PRODUKT-SCHNITTSTELLEN

6.2.1 DAS ATEO-SYSTEM

Die Komponente SAM ist hierbei ein Bestandteil des ATEO-Systems, welches zudem aus den Komponenten **ATEO Master Display** (AMD) und **Automatiken** (AM) besteht (siehe *Abbildung 11*).

SAMs übernimmt hierbei die Simulation des Tracking-Prozesses sowie die Protokollierung der während des Versuchs anfallenden Daten (für weitere Details siehe *Abschnitt 3*).

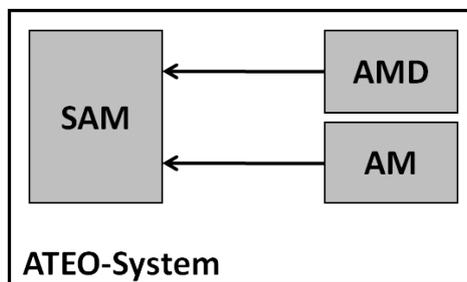


Abbildung 11 - Das ATEO-System mit seinen Komponenten SAM, AMD und AM, wobei die Pfeile die Zugriffsrichtung angeben.

Der AMD ermöglicht es dem Operateur komfortabel und übersichtlich den Tracking-Prozess zu beobachten und zudem Eingriffe zu tätigen, falls dies nötig sein sollte. Für die Überwachung des Tracking-Prozesses kommen eine Streckenansicht einschließlich des Tracking-Objekts, ein Videobild der MWB und eine Darstellung der Joystickeingaben zum Einsatz. Darüber hinaus werden Leistungsdaten wie die Genauigkeit und die Geschwindigkeit des Trackings integriert am Tracking-Objekt in Form eines Schweifes dargestellt. Der Operateur hat zudem die Möglichkeit auditive wie visuelle Hinweise an die MWB auszugeben oder gar in den Steuerungsprozess einzugreifen, indem er die maximale Geschwindigkeit festlegt, das Lenken in eine bestimmte Richtung blockiert oder der Steuerungsanteil der MWB manipuliert. Weitere Details zum AMD sind in [6] nachzulesen.

Die AM besitzen hierbei die gleiche Aufgabe wie der Operateur, nämlich das Tracking zu überwachen und zu führen. Allerdings werden sie von den Entwicklern des Systems entworfen und implementiert. Die Qualität der AM und deren Wirkung auf die Prozessüberwachung und -führung stellt ein Leistungsmaß hinsichtlich der Arbeit der Entwickler dar. Somit ist es möglich die Leistungen des Operateurs mit den Leistungen der Entwickler zu vergleichen. Mit diesem Vergleich ist es dann möglich zu untersuchen, ob eine bestimmte Aufgabe durch einen Operateur bewältigt werden sollte oder ob es günstiger ist, für diese Aufgabe ein AM durch die Entwickler konzipieren und implementieren zu lassen. Darüber hinaus wird so eine Optimierung der jeweiligen Leistung in der Prozessüberwachung und -führung möglich. Die Details zu den AM werden in [7] ausführlich beschrieben.

³ Weitere Informationen zu Squeak sind auf www.squeak.org und in [11] verfügbar.

6.2.2 VERSUCHSAUFBAU

Der Versuchsaufbau ist in *Abbildung 12* zu sehen. Der Aufbau erstreckt sich über zwei Räume, was durch einen senkrechten Strich angedeutet wird. Im ersten Raum befinden sich die MWB, welche die Tracking-Aufgabe absolvieren. SAMs wird hierbei auf dem Rechner *ATEO 1* ausgeführt. Für die Interaktion mit SAMs stehen ein bis zwei Joysticks und eine Maus zur Verfügung, wobei die Joysticks zur Steuerung des Tracking-Objekts eingesetzt werden und die Maus zum Quittieren von Anweisungen (MWB) sowie dem Konfigurieren einiger Versuchsparameter (Versuchsleiter). Für die Darstellung wird ein 19" Monitor benutzt. Darüber hinaus wird mittels einer Webcam ein Videobild der MWB an den Rechner *ATEO 2* übertragen, welcher sich im zweiten Raum befindet. Darüber hinaus werden 2 Laptops verwendet, um die MWB mit ihrer Zielstellung vertraut zu machen, welche jedoch nicht in *Abbildung 12* zu sehen sind.

Auf dem Rechner *ATEO 2* wird die ATEO-Komponente AMD ausgeführt (siehe *Abschnitt 6.2.1*), welche u.a. auch das übertragene Videobild der MWB für den Operateur darstellt. Die Webcam ist hierbei direkt durch ein Kabel an *ATEO 2* angeschlossen. Zudem sind die Rechner *ATEO 1* und *ATEO 2* durch einen Ethernet-Hub vernetzt, so dass die Komponenten SAMs und AMD miteinander kommunizieren können. Weiterhin stehen dem Operateur ein 30" Monitor, eine Tastatur und eine Maus zur Bedienung des AMD zur Verfügung. Eine zweite Webcam überträgt das Videobild des Operateurs an einen weiteren Rechner: *ATEO 3* (aus Performance-Gründen).

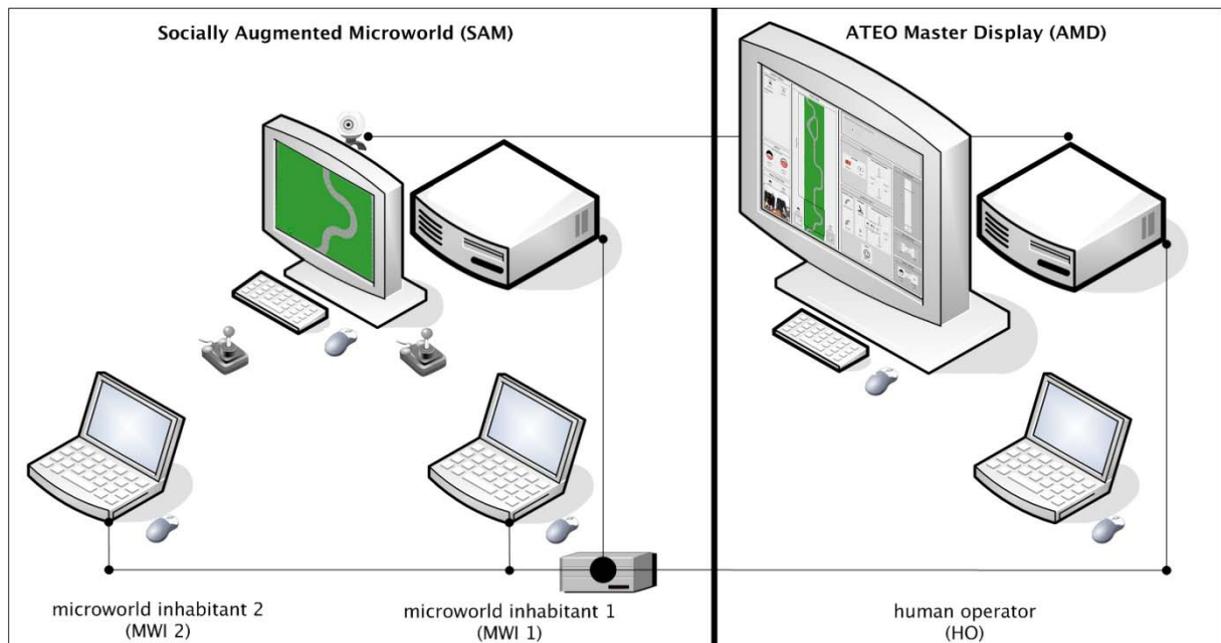


Abbildung 12 - Versuchsaufbau der psychologischen Untersuchungen.

7 LITERATUR

- [1] BALZERT, HELMUT: *Lehrbuch der Software-Technik, Bd. 1 Software-Entwicklung*, Spektrum Akademischer Verlag, 2000. – ISBN 3-82740480-0
- [2] LIFE CYCLE DATA HARMONIZATION WORKING GROUP: IEEE Std 830-1998, *Recommended Practice for Software Requirements Specifications*. IEEE, 1998.
- [3] NACHTWEI, JENS ; KAIN, SASKIA ; KRINNER, CORDULA ; GROSS, BARBARA: *ATEO – ein Beitrag zum Graduiertenkolleg prometei*, 2008. URL: <http://www.psychologie.hu-berlin.de/prof/ingpsy/forschung/ateo> (abgefragt am 07.10.2008).
- [4] GROSS, BARBARA ; NACHTWEI, JENS: *Using the microworld to acquire knowledge for developers and operators*. In: WAARD, D. DE (Hrsg.) ; HOCKEY, G.R.J. (Hrsg.) ; NICKEL, P. (Hrsg.) ; BROOKHUIS, K.A. (Hrsg.): *Human Factors Issues in Complex System Performance*. Maastricht, the Netherlands : Shaker Publishing, 2007, S. 345-350.
- [5] WANDKE, HARTMUT ; NACHTWEI, JENS: *The different human factor in automation: the developer behind versus the operator in action*. In WAARD, D. DE (Hrsg.) ; FLEMISCH, F. O. (Hrsg.) ; LORENZ, B. (Hrsg.) ; OBERHEID, H. (Hrsg.) ; BROOKHUIS, K.A. (Hrsg.): *Human factors for assistance and automation*. Maastricht, the Netherlands : Shaker Publishing, 2008, S. 493-502.
- [6] SCHWARZ, HERMANN: *Fenster zum Prozess: ein Operateursarbeitsplatz zur Überwachung und Kontrolle von kooperativem Tracking*, Humboldt-Universität zu Berlin, Institut für Informatik, Unveröffentlichte Diplomarbeit, 2009.
- [7] KESSELRING, KAI: *Entwicklung einer Softwarekomponente zur Systemprozessüberwachung in einer psychologischen Versuchsumgebung*, Humboldt-Universität zu Berlin, Institut für Informatik, Unveröffentlichte Diplomarbeit, 2009.
- [8] NIESTROJ, NICOLAS: *Vernetzung im ATEO Projekt aus inhaltlicher und technischer Sicht*, Humboldt-Universität zu Berlin, Institut für Informatik, Unveröffentlichte Diplomarbeit, 2009.
- [9] Übersicht Spielbretter und Rot-Werte, Stand 31.12.07
- [10] BLACK, ANDREW ; DUCASSE, STÉPHANE ; NIERSTRASZ, OSCAR ; POLLET, DAMIEN ; CASSOU, DAMIEN ; DENKER, MARCUS: *Squeak by Example*, Square Bracket Associates, 2008. – URL <http://SqueakByExample.org/index.html>. – ISBN 978-3-9523341-0-2
- [11] GROSS, BARBARA ; SACKLOWSKI, ULRICH: *ATEO-System – Nutzerdokumentation*, 2008

8 QUELLEN

Im Rahmen des ATEO-Projektes wurden verschiedene Artefakte erstellt, welche über die folgenden Webseiten verfügbar sind:

- <https://www.assembla.com/spaces/ATEO>
Die Projekt-Seite von ATEO. Hier befinden sich ein GIT-Repository und ein Entwickler-Wiki. Ersteres enthält hierbei sowohl die aktuellen Quelltexte der ATEO-Komponenten als auch diverse Dokumente.
- <http://zope.psychologie.hu-berlin.de/prof/ingpsy/forschung/ateo>
Diverse Informationen zum Projekt ATEO aus psychologischer Sicht.

9 GLOSSAR

| | |
|-------------------|--|
| Assistenzsystem | Ein System, das an der Überwachung eines Prozesses beteiligt ist, jedoch den Operateur beim Erreichen von bestimmten Grenzwerten lediglich warnt oder aber selbst regulierend eingreift. |
| Automatik | Ein System, welches selbstständig die Überwachung und / oder Steuerung eines Prozesses übernimmt, ohne dass ein Mensch eingreift. Lediglich bei Notfällen kann es zu einem Eingreifen durch einen Menschen kommen. |
| CSV | Das Dateiformat CSV beschreibt den Aufbau einer Textdatei zur Speicherung oder zum Austausch einfach strukturierter Daten. Die Abkürzung CSV steht für Character Seperated Values. |
| Interface | Graphische Benutzeroberfläche, auch Graphics User Interface (GUI) genannt |
| Logging | Das Aufzeichnen aller versuchsrelevanten Daten, die nötig sind um die gewünschte Versuchsauswertung vorzunehmen. |
| Mikrowelt | Abstraktion von der realen Welt, welche benutzt wird, um reale dynamische und komplexe Prozesse zu simulieren. Und es so ermöglicht, das Verhalten von Operateuren oder Automaten unter kontrollierten Laborbedingungen zu untersuchen. |
| Mikroweltbewohner | Eine Versuchsperson im System ATEO, welche die Aufgabe hat das Fahrzeug (Tracking-Objekt) zu steuern. |
| Operateur | Eine Person oder/und ein Softwaresystem, die in den Versuchsablauf korrigierend, assistierend eingreifen kann, z.B. Hinweise zum Streckenverlauf geben, Eingriff in die Steuerung. |
| Steuerungsanteil | Der Steuerungsanteil gibt an, wie stark sich die Steuereingaben eines MWB auf die neue Position des Tracking-Objekts auswirken. Die Verteilung der Steuergewalt wird hierbei prozentual vorgenommen. Besitzen beispielsweise beide MWB eine Steuergewalt von 50%, so gehen die Steuereingaben beider MWB zu gleichen Teilen ein. |
| Team | Eine Gruppe von Mikroweltbewohnern, welche gemeinsam das Tracking-Objekt steuern. |
| Tracking | Ein Tracking bezeichnet hier die Aufgabe, ein Objekt in einem bestimmten Bereich (z.B. einer Straße) zu halten (Spurhaltungsaufgabe). |
| Versuchsgruppe | Eine Gruppe von Versuchspersonen, welche an einem Versuch unter bestimmten Bedingungen teilnehmen. |
| Versuchsleiter | Person, die für die Leitung eines Versuches zuständig ist. |

Versuchsperson, Proband Person, die an einem Versuch teilnimmt. Das Verhalten dieser Personen unter bestimmten Bedingungen ist der Untersuchungsgegenstand eines Versuchs.

10 ANHANG: IMPLEMENTATIONSDetails

10.1 STRECKENANIMATION

Die Tracking-Simulation geschieht in mehreren gleichartigen Simulationsschritten, welche wiederholt ausgeführt werden bis das Ende der Tracking-Simulation erreicht ist. Ein Simulationsschritt wird alle 30 bis 49 ms ausgeführt (siehe *Abschnitt 2*). In einem solchen Schritt werden die Steuereingaben der MWB abgefragt, die neue Position des Tracking-Objektes, der Strecke und den Hindernissen ausgerechnet und auf dem Bildschirm gezeichnet. Darüber hinaus werden die Eingaben durch den Operateur oder die Automaten umgesetzt. Sollte es zu Kollisionen kommen, so werden auch diese behandelt.

Durch das Verschieben des Streckenbildes in den einzelnen Simulationsschritten entsteht der Eindruck einer Fahrt.

Hierbei wird das Streckenbild um den Betrag von $ypos_{T0}^A$ (siehe *Abschnitt 4.2.2*) verschoben. Die Y-Koordinate des Tracking-Objektes auf dem Bildschirm verändert sich hierbei nicht. Dennoch ändert sich die Y-Koordinate relativ zur Gesamtstrecke, da das Streckenbild „unter“ dem Tracking-Objekt bewegt wird.

11 ABWEICHUNGEN UND FEHLER

11.1 VERSUCHSSTEUERUNG

In SAMs 2.2 kann der Versuch *nur noch* via *Alt + .* abgebrochen werden und dann komplett von vorn gestartet werden. Eine Steuerung des Versuchsablaufes ist demnach nur noch rudimentär möglich. Dies wird durch das verwendete *BitBlitting* verursacht.