EILERS, M. & MÖBUS, C., *Lernen eines modularen Bayesian Autonomous Driver Mixture-of-Behaviors (BAD MoB) Modells*, in H. Kolrep & Th. Jürgensohn, (Hrsg.), Fahrermodellierung - Zwischen kinematischen Menschmodellen und dynamisch-kognitiven Verhaltensmodellen, Fortschrittsbericht des VDI in der Reihe 22, Nr.32, S. 61 - 74, Düsseldorf: VDI-Verlag, 2010, ISBN 978-3-18-303222-8

Lernen eines modularen Bayesian Autonomous Driver Mixture-of-Behaviors (BAD MoB) Modells

M. Eilers[1,2], C. Möbus[1,2]

OFFIS e.V. / C.v.O. Universität, Oldenburg

Das Human Centered Design (HCD) intelligenter Transportsysteme erfordert Computermodelle menschlichen Verhaltens und menschlicher Kognition, die als Fahrermodelle in Verkehrssimulationen verwendet werden können. Die gängige Vorgehensweise bei ihrer Entwicklung ist die Entwicklung von Softwareprototypen auf der Basis der Regelungstheorie oder regelbasierter Ansätze der Künstlichen Intelligenz. Diese Prototypen werden dann im Nachhinein auf Usability und empirische Validität überprüft. Im Gegensatz dazu schlagen wir eine Alternative auf Basis des maschinellen Lernens und der Stochastik vor: die Schätzung Bayesscher Fahrermodelle aus Verhaltensdaten. Die resultierenden Bayesian Autonomous Driver (BAD) Models sind im Prinzip empirisch valide, wenn die in den Daten enthaltenen statistischen Relationen und bedingten Unabhängigkeiten schon bei der Konstruktion berücksichtigt wurden. BAD Modelle können so modelliert werden, dass sie in der Lage sind, komplexe Fähigkeiten in einfachere Basisfähigkeiten zu zerlegen, oder umgekehrt: BAD Mixture-of-Behaviors (BAD MoB) Modelle. Hier wird die effiziente Implementierung eines solchen BAD MoB Modells beschrieben. Das Modell wird zur autonomen Echtzeitsteuerung eines simulierten Fahrzeugs eingesetzt. Es ist in der Lage, komplexes Verhalten verschiedener Kompetenzlevel (Levels of Expertise) durch das Mischen und Aneinanderreihen einfacherer Verhaltensweisen zu erzeugen.

1 Einleitung

Kompetenzen und Lernprozesse eines Fahrers können durch ein Drei-Ebenen-Modell mit einer *kognitiven*, einer *assoziativen* und einer *autonomen* Ebene beschrieben werden (Fitts und Posner, 1967; Anderson, 2002). Für jede Ebene haben sich dabei spezielle Modellierungsansätze eingebürgert: *Produktionssysteme* für die kognitive und assoziative Ebene, *kontrolltheoretische* und *probabilistische* Modelle für die autonome Ebene. Aufgrund der großen Variabilität menschlichen Verhaltens und der *prinzipiell* nur unzureichenden Kenntnis menschlicher Kognition und Unsicherheit über die uns umgebende dynamische Welt (Bessière et al., 2008), konzeptualisieren, entwickeln und implementieren wir menschliche Fahrermodelle in Form von probabilistischen Modellen. Erste Schritte zur Modellierung der *lateralen* und *longitudinalen* Kontrolle von menschlichen Fahrern durch *statische* BAD Modelle finden sich in Möbus und Eilers (2008) und durch *dynamische* Bayessche Sensor-Motor Modelle in Möbus et al. (2009a, 2009b). Hier entwickeln wir ein dynamisches *modulares* BAD MoB Modell, das in der Lage ist, komplexes Verhalten in einfachere Verhalten zu zerlegen oder vice versa. Das Modell ermöglicht die Implementierung einer Bibliothek von einfachen

[1] project Integrated Modeling for Safe Transportation (IMOST) sponsored by the Government of Lower Saxony, Germany under contracts ZN2245, ZN2253, ZN2366

Sensor-Motor Schemata (Behaviors). Durch Mischen und Aneinanderreihen solcher reinen Basis-Behaviors kann dann situationsbedingtes und komplexes Verhalten erzeugt werden.

1.1 Bayessche Programme

BAD MoB Modelle sind beeinflusst von probabilistischen Expertensystemen (Pearl, 2009) und Konzepten des Bayesian (Robot) Programmings (Bessiere et al., 2003, 2008, Lebeltel et al., 2004). Bayessche Programme (BP) spezifizieren gemeinsame Wahrscheinlichkeitsverteilungen von Zufallsvariablen in das Produkt einfacherer bedingter Wahrscheinlichkeitsverteilungen. Durch ein Bayessches Programm ist es möglich, ein Fahrermodell zu entwickeln, das ein (virtuelles) Fahrzeug autonom in Echtzeit steuert. Die Struktur eines Bayesschen Programms ist in Bild 1 dargestellt.

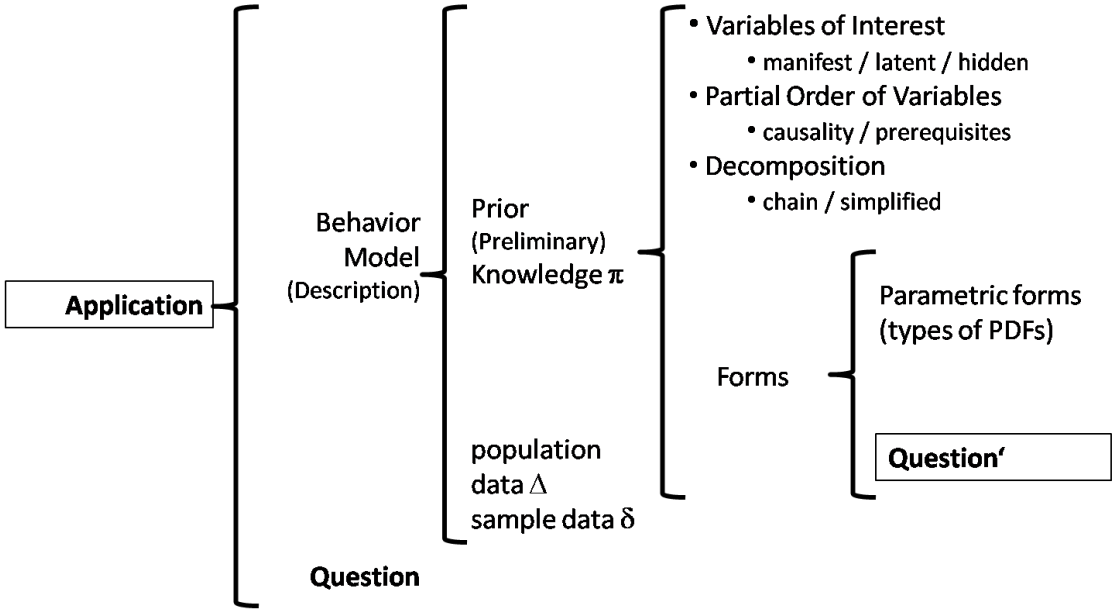

Bild 1: Struktur eines Bayesschen Programms (adaptiert von Bessiere et al., (2003, 2008), Lebeltel et al., (2004)).

1.2 Bayesian Autonomous Driver Mixture-of-Behaviors Modelle

In Möbus und Eilers (2010a) haben wir eine probabilistische Architektur für Modelle verschiedener menschlicher Kompetenzlevels präsentiert. Sie ermöglichen es, einfache Verhaltensweisen (Behaviors) in komplexeren Fahrmanövern (Maneuvers) eventuell gewichtet wieder zu verwenden. Diese Modelle implementieren das Sensor-Motor-System eines menschlichen Fahrers in einer psychologisch motivierten Mixture-of-Behaviors Architektur mit autonomer und zielbasierter Aufmerksamkeitssteuerung. Mittels eines Proof of Concepts auf der Basis von künstlichen aber plausiblen Daten und ersten Modellierungsversuchen mit echten im Simulator erhobenen Fahrdaten haben wir zeigen können, dass BAD MoB Modelle die Fähigkeit besitzen Verhalten zu mischen, vorherzusagen, gefährliche Situationen auf Vorgängersituationen zurückzurechnen (Abduktion) und antizipatorische Pläne und Kontrolle zu generieren.

2 Modulare BAD MoB Modelle

Mit zunehmender Anzahl von modellierten Aktions- und Perzept-Variablen und insbesondere von latenten Zustands- und Behavior-Variablen, können BAD MoB Modelle schnell zu komplex für eine Echtzeitsteuerung von (simulierten) Fahrzeugen werden, wenn sie nur als einziges unstrukturiertes dynamisches Bayesnetz konzipiert werden, wie das z.B. in Möbus und Eilers (2010a) geschehen ist.

Wir schlagen daher als eine effektive Möglichkeit zur Modellierung von BAD MoB Modellen die *Description* oder *Behavior Combination* (Bessière et al., 2003; 2008, p.36f) vor, bei der jedes gewünschte *Behavior* durch ein *eigenes* Sensor-Motor-Schema in Form eines separaten und eigenständigen Bayesschen Programm realisiert wird (Eilers und Möbus, 2010). Diese Bayesschen Programme bilden eigenständige Module, die sich zu immer komplexeren Verhalten mittels gewichteter Summen kombinieren lassen. Wir bezeichnen diese Modelle daher als *modulare* BAD MoB Modelle.

2.1 Grundlagen

2.1.1 Spezifikation einer Verhaltenshierarchie

Die Grundlage eines modularen BAD MoB Modells ist die sukzessive Verfeinerung des gewünschten Gesamtverhaltens in immer einfachere Verhalten. Eine solche Verfeinerung bildet eine Verhaltensbibliothek in Form einer Verhaltenshierarchie. Die Ebenen oder Kompetenzlevel dieser Verhaltenshierarchie reichen dabei vom komplexen bis zum einfachen Verhalten. Wir betrachten das Kompetenzlevel der Fahrszenarien (*Scenarios*), der Fahrmanöver (*Maneuvers*) und der einfachen Teilmanöver, die wir als Fahrverhalten oder *Behaviors* bezeichnen. Ein Fahrszenario (Scenario) lässt sich in eine Reihe möglichst orthogonaler Fahrmanöver (Maneuvers) verfeinern, die sich ihrerseits weiter in eine Reihe von möglichst orthogonalen Fahrverhalten (Behaviors) zergliedern lässt.

Als Beispiel zeigt Bild 2 eine allgemeine Verhaltenshierarchie über drei Kompetenzlevel. Ein einzelnes Scenario $Scenario_i$ wird zunächst in eine Anzahl von m Manöver $Maneuver_j, j = 1, \dots, m$ und jedes $Maneuver_j$ weiter in eine Anzahl n_j Behaviors $Behavior_{j_k}, k = 1, \dots, n_j$ verfeinert[2]. Für jedes $Maneuver_j$ gibt es ein entsprechendes *Gating*-Modell $\mathcal{G}_j$, das seinerseits ein *Behavior-Classification*-Modell $\mathcal{B}_j$ und eine Anzahl von n_j Action-(Classification-) Modellen $\mathcal{A}_{j_k}, k = 1, \dots, n_j$ enthält.

2.1.2 Spezifikation von interessierenden Variablen

Zur Implementierung eines BAD MoB Modells müssen alle interessierenden Variablen spezifiziert werden, die bei der Modellierung benötigt werden oder werden könnten. Dies sind zunächst alle *Aktions*-Variablen durch die sich das gewünschte Gesamtverhalten äußert. Sie sollen im Folgenden als A bezeichnet werden. A ist im Falle der BAD MoB Modelle für alle Komponenten (Behaviors) gleich und repräsentiert die konkreten Fahraktionen der Lateral- und Longitudinalkontrolle wie Lenkwinkel, Gas- und Bremspedalstellungen.

[2] Es gilt zu beachten, dass nicht notwendigerweise $n_j = n_{j'}$ gilt, wenn $j \neq j'$.

Hinzu kommen die *Perzept*-Variablen $P_1, \dots, P_P$. Die Konjunktion aller verfügbaren Perzept-Variablen eines modularen BAD MoB Modells soll im Folgenden als P_{All} bezeichnet werden.

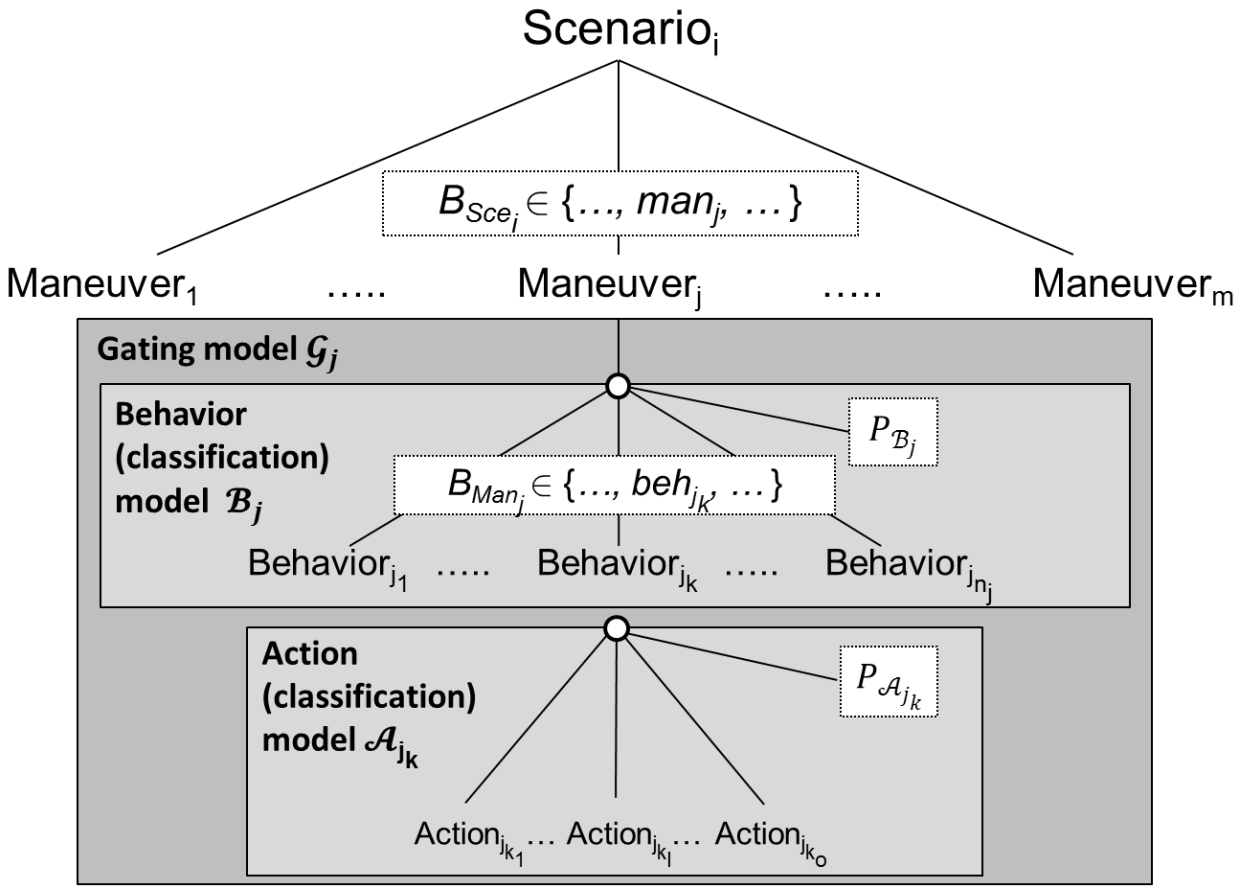

Bild 2: Verhaltenshierarchie über die drei Kompetenzlevel der (Driving) Scenarios, (Driving) Maneuvers und (Driving) Behaviors mit dazugehörigen geschachtelten Geltungsbereichen der Gating-, Behavior-Classification- und Action-Classification-Modelle.

Zur Repräsentation der verschiedenen Verhalten einer Verhaltenshierarche werden zusätzlich eine Menge von *Behavior*-Variablen benötigt. Behavior-Variablen sind nominale (qualitative) Variablen und ihre Anzahl und jeweilige Kardinalität ergibt sich direkt aus der intendierten Verhaltenshierarchie. Jedes komplexe Verhalten, das sich in einfachere Verhalten verfeinern lässt, wird durch eine Behavior-Variable repräsentiert, deren Werte die einfacheren Verhalten repräsentieren. Für die in Bild 2 dargestellte Verhaltenshierarchie werden die zu dem Szenario $Scenario_i$ gehörenden Manöver $Maneuver_j, j = 1, \dots, m$ durch eine Behavior-Variable B_{Sce_i} mit m verschiedenen Werten $man_j, j = 1, \dots, m$ repräsentiert. In gleicher Weise werden die zu einem Manöver $Maneuver_j, j = 1, \dots, m$ gehörenden n_j Behaviors $Behavior_{j_k}, k = 1, \dots, n_j$ durch Behavior-Variablen B_{Man_j} mit n_j verschiedenen Werten $beh_{j_k}, k = 1, \dots, n_j$ repräsentiert (vergl. Bild 2).

Auf die Verwendung von möglichen (latenten) *State*-Variablen, die man z.B. für kognitive Karten benötigt, wird bislang noch verzichtet (s.a. Möbus und Eilers, 2010a, b).

2.1.3 Experimentelle Lerndaten

Modulare BAD MoB Modelle sollen in der Tradition von BAD Modellen möglichst vollständig aus empirischen Daten gelernt werden. Eine Voraussetzung für die Entwicklung eines modularen BAD MoB Modells ist daher eine Datenbasis δ (vergl. Bild 1) von experimentellen Lerndaten in Form von multivariaten Zeitreihen. Die Zeitreihen werden in Experimenten, die alle zu modellierenden Verhalten abdecken, aufgezeichnet. Um die verschiedenen als Reaktion gezeigten Verhalten in den experimentellen Lerndaten δ identifizieren zu können, muss δ hinsichtlich des Verhaltens annotiert werden. Dies geschieht zum jetzigen Zeitpunkt mittels einer nachträglichen Analyse der Lerndaten und einer manuellen Annotation. Dabei muss für jede Zeile der experimentellen Lerndaten δ jeweils das aktuell aktive $Maneuver_j$, $j = 1, \dots, m$ und weiter das aktuell aktive $Behavior_{j_k}, k = 1, \dots, n_j$ angegeben werden.

2.2 Modellierung eines modularen BAD MoB Modells

Als Beispiel zur Entwicklung eines modularen BAD MoB Modells und zur Veranschaulichung der Funktionsweise desselben, soll die in Bild 2 enthaltene reduzierte Verhaltensbibliothek über ein einzelnes $Maneuver_j$ dienen. Zur Vereinfachung werden die Teilmodelle des modularen BAD MoB Modells als *statische* Modelle konzipiert; das Prinzip ist jedoch ohne große Änderungen auf *dynamische* Modelle übertragbar.

Zur Implementierung eines modularen BAD MoB Modells wird zunächst jedes Behavior einer Verhaltenshierarchie durch ein eigenes separates Sensor-Motor-Schema in Form eines Bayesschen Programms modelliert, das wir als ein *Action*-Model bezeichnen. Für die in Bild 2 gezeigte Verhaltenshierarchie müssen wir für jedes $Behavior_{j_k}, k = 1, \dots, n_j$ jeweils n_j *Action*-Modelle $\mathcal{A}_{j_k}, k = 1, \dots, n_j$ erstellen. Das Ergebnis ist eine Bibliothek von *Action*-Modellen $\mathcal{A}_{j_k}, k = 1, \dots, n_j;\ j = 1, \dots, m$.

2.2.1 *Action*-Modelle

Jedes *Action*-Modell dient der Modellierung eines einzelnen Behaviors und soll verwendet werden, um Fahraktionen des jeweiligen Verhaltens vorherzusagen bzw. vorzuschlagen. Dafür benötigt man zum einen die Aktions-Variablen A, zum anderen eine Auswahl an Perzept-Variablen aus der Menge P_{All}. Um ein *Action*-Modell für eine effiziente Inferenz so einfach wie möglich zu halten, sollen aus der Menge P_{All} nur die zur Modellierung des gewünschten Verhaltens unbedingt benötigten *cues*, die sogenannten *Peephole*-Perzepte berücksichtigt werden.

Angenommen, das Action-Modell $\mathcal{A}_{j_k}$ benötigt für die Modellierung des $Behavior_{j_k}$ nur die Peephole-Perzepte $P_{\mathcal{A}_{j_k}} = \left\{P_{j_{k_1}}, \dots, P_{j_{k_p}}\right\}, P_{\mathcal{A}_{j_k}} \subseteq P_{All}$, das Vorwissen bzw. den Kontext $\pi_{\mathcal{A}_{j_k}}$ und die mit dem intendierten Verhalten $Behavior_{j_k}$ annotierten empirischen Lerndaten $\delta_{\mathcal{A}_{j_k}}$, dann ist die vollständige gemeinsame Verteilung definiert als:

$$P\left(A, P_{\mathcal{A}_{j_k}} \mid \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right) = P\left(A, P_{j_{k_1}}, \dots, P_{j_{k_p}} \mid \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right).$$

Die grundlegende Struktur der Dekomposition der vollständigen gemeinsamen Verteilung von *Action*-Modellen konzipieren wir als naiven Bayessche Klassifizierer. Für die Dekomposition des *Action*-Modells $\mathcal{A}_{j_k}$ folgt daher:

$$(1)\ P\left(A, P_{\mathcal{A}_{j_k}} \mid \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right) = P\left(P_{\mathcal{A}_{j_k}} \mid A,\ \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right) \cdot P\left(A \mid \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right)$$

$$= P\left(A \mid \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right) \cdot \prod_{l=1}^{p} P\left(P_{j_{k_l}} \mid A, \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right).$$

Lernen von Action-Modellen

Die zur Modellierung des intendierten Verhaltens notwendigen Peephole-Perzepte $P_{\mathcal{A}_{j_k}}$ werden mit Methoden des maschinellen Lernens aus den experimentellen Daten δ gelernt. Dafür wird eine heuristische Suche im Raum der möglichen Kombinationen der insgesamt verfügbaren Perzept-Variablen P_{All} durchgeführt, wobei jede untersuchte Kombination durch das von Schwartz (1978) entwickelte *Bayesian Information Criterion* (*BIC*) als Gütemaß bewertet wird. Das BIC belohnt die Vorhersagegüte des resultierenden Modells, besitzt jedoch einen bestrafenden Faktor für die Anzahl der dabei verwendeten Perzept-Variablen. Die heuristische Suche mit dem BIC führt zu einer zumindest lokal optimalen Kombination an Perzept-Variablen, die dann als Peephole-Perzepte eines *Action*-Modells verwendet werden.

Verwendungszweck

Nach dem Lernen der notwendigen Peephole-Perzepte $P_{\mathcal{A}_{j_k}}$ sollte das *Action*-Modell in der Lage sein, gemäß dem intendierten Verhalten Fahraktionen für die durch die Peephole-Perzepte $P_{\mathcal{A}_{j_k}}$ definierte Situation vorherzusagen. Dafür wird die folgende Frage definiert:

$$(2)\ P\left(A \mid P_{\mathcal{A}_{j_k}}, \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right).$$

2.2.2 *Behavior-Classification*-Modelle

Für die Erzeugung situationsadäquaten Verhaltens muss entschieden werden, ob ein einfaches Basisverhalten ausreicht oder ob eine bzw. welche Mischung von Basisverhalten zur Erzeugung des gewünschten komplexen Verhaltens benötigt wird. Diese Vorhersage wird durch eine weitere Form von Bayesschen Programmen realisiert, die wir als *Behavior-Classification*-Modelle bezeichnen. Die Definition eines derartigen Modells erfolgt exemplarisch am Beispiel des *Behavior-Classification*-Modells $\mathcal{B}_j$. Es dient der Inferenz situationsangepassten Verhaltens $Maneuver_j$ als Mischung der n_j Behaviors $Behavior_{j_k}, k = 1, \ldots, n_j$.

Als interessierende Variablen benötigt das *Behavior-Classification*-Modell $\mathcal{B}_j$ die Behavior-Variable B_{Man_j} und eine eigene Auswahl von zur Identifikation des richtigen Verhaltens notwendigen *Peephole*-Perzepten $P_{\mathcal{B}_j} = \left\{P_{j_1}, \ldots, P_{j_q}\right\}, P_{\mathcal{B}_j} \subseteq P_{All}$. Werden die verfügbaren Lerndaten als $\delta_{\mathcal{B}_j}$ und das Vorwissen als $\pi_{\mathcal{B}_j}$ bezeichnet, dann ist die vollständige gemeinsame Verteilung des *Behavior-Classification*-Modells $\mathcal{B}_j$ definiert als:

$$P\left(B_{Man_j}, P_{\mathcal{B}_j} \mid \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right) = P\left(B_{Man_j},\ P_{j_1}, \ldots, P_{j_q} \mid \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right).$$

Die grundlegende Struktur der Dekomposition der vollständigen gemeinsamen Verteilung von *Behavior-Classification*-Modellen wird wiederum als naiver Bayesscher Klassifizierer konzipiert, für die Dekomposition des *Behavior-Classification*-Modells $\mathcal{B}_j$ folgt daher:

$$(3)\ P\left(B_{Man_j}, P_{\mathcal{B}_j} | \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right) = P\left(P_{\mathcal{B}_j} | B_{Man_j}, \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right) \cdot P\left(B_{Man_j} | \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right)$$

$$= P\left(B_{Man_j} | \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right) \cdot \prod_{l=1}^{q} P\left(P_{j_l} \mid B_{Man_j}, \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right).$$

Lernen von Behavior-Classification-Modellen

Das Lernen der zur Modellierung eines *Behavior-Classification*-Modells benötigten Peephole-Perzepte erfolgt analog zu *Action*-Modellen durch eine heuristische Suche im Raum der möglichen Kombinationen von Peephole-Perzepten.

Verwendungszweck

Nach dem Lernen der notwendigen Peephole-Perzepte $P_{\mathcal{B}_j}$ sollte das *Behavior-Classification*-Modell $\mathcal{B}_j$ in der Lage sein, die zur Bewältigung einer durch die Peephole-Perzepte $P_{\mathcal{B}_j}$ definierten Situation adäquate Mischung der Behaviors zu bestimmen. Dafür wird die folgende Frage definiert:

$$(4)\ P\left(B_{Man_j} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right).$$

2.2.3 *Gating*-Modelle

Mittels einer als *Description* oder *Behavior Combination* (Bessière et al., 2003, 2008) bezeichneten Technik der Bayesschen Programmierung lassen sich ein *Behavior-Classification*-Modell und eine Anzahl von *Action*-Modellen durch eine dritte Art von Bayesschen Programm zu einem Gesamtmodell kombinieren. Das ermöglicht dann, komplexes Verhalten durch ein *Mixture-of-Behaviors* zu erzeugen. Diese dritte Art von Bayesschen Programmen bezeichnen wir als *Gating*-Modelle. Die Definition eines *Gating*-Modells erfolgt exemplarisch am Beispiel eines *Gating*-Modells $\mathcal{G}_j$ zur Kombination des *Behavior-Classification*-Modells $\mathcal{B}_j$ mit den n_j *Action*-Modellen $\mathcal{A}_{j_k}, k = 1, \dots, n_j$.

Gating-Modelle müssen weder gelernt noch „von Hand“ modelliert werden. Die Spezifikation eines *Gating*-Modells erfolgt stattdessen generisch und kann automatisiert werden. Das *Gating*-Modell $\mathcal{G}_j$ dient dabei nur der Verknüpfung des *Behavior-Classification*-Modells $\mathcal{B}_j$ mit den n_j *Action*-Modellen $\mathcal{A}_{j_k}, k = 1, \dots, n_j$. Es soll kein weiteres Wissen modelliert werden; folglich benötigt das *Gating*-Modell auch keine Lerndaten. Interessierende Variablen eines *Gating*-Modells sind alle die Variablen, die zum Befragen des *Behavior-Classification*-Modells und der *Action*-Modelle benötigt werden. Im Falle des *Gating*-Modells $\mathcal{G}_j$ sind dies die Behavior-Variable B_{Man_j} und die Peephole-Perzepte $P_{\mathcal{B}_j}$ des *Behavior-Classification*-Modells $\mathcal{B}_j$, sowie die Aktions-Variablen A und alle benötigten Peephole-Perzepte $P_{\mathcal{A}_j} = \{P_{\mathcal{A}_{j_k}}, k = 1, \dots, n_j\}$ der n_j *Action*-Modelle $\mathcal{A}_{j_k}, k = 1, \dots, n_j$. Wird das Vorwissen des *Gating*-Modells als $\pi_{\mathcal{G}_j}$ bezeichnet, ist die vollständige gemeinsame Verteilung definiert als:

$$P\left(A, B_{Man_j}, P_{\mathcal{A}_j}, P_{\mathcal{B}_j} \mid \pi_{\mathcal{G}_j}\right).$$

Die Dekomposition des *Gating*-Modells $\mathcal{G}_j$ ergibt sich als:

$$(5)\ P\left(A, B_{Man_j}, P_{\mathcal{A}_j}, P_{\mathcal{B}_j} \mid \pi_{\mathcal{G}_j}\right) =$$
$$P\left(P_{\mathcal{A}_j}, P_{\mathcal{B}_j} \mid \pi_{\mathcal{G}_j}\right) \cdot P\left(B_{Man_j} \mid P_{\mathcal{A}_j}, P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right) \cdot P\left(A \mid B_{Man_j}, P_{\mathcal{A}_j}, P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right).$$

Sie kann unter den Unabhängigkeitsannahmen $\left(B_{Man_j} \perp P_{\mathcal{A}_j} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right)$ und $(A \perp P_{\mathcal{B}_j} \mid B_{Man_j}, P_{\mathcal{A}_j}, \pi_{\mathcal{G}_j})$ weiter vereinfacht werden zu:

$$(6.1)\ P\left(A, B_{Man_j}, P_{\mathcal{A}_j}, P_{\mathcal{B}_j} \mid \pi_{\mathcal{G}_j}\right) =$$
$$P\left(P_{\mathcal{A}_j}, P_{\mathcal{B}_j} \mid \pi_{\mathcal{G}_j}\right) \cdot P\left(B_{Man_j} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right) \cdot P\left(A \mid B_{Man_j}, P_{\mathcal{A}_j}, \pi_{\mathcal{G}_j}\right).$$

Die Dekomposition besteht aus drei Termen. Der erste Term $P\left(P_{\mathcal{A}_j}, P_{\mathcal{B}_j} \mid \pi_{\mathcal{G}_j}\right)$ beschreibt eine a-priori Verteilung über alle Peephole-Perzepte. Bezogen auf die spätere Verwendung des *Gating*-Modells stellt er einen Normalisierungsfaktor dar und kann als gleichverteilt angenommen werden. Der zweite Term $P\left(B_{Man_j} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right)$ bezeichnet die bedingte Verteilung der Behaviors, d.h. die adäquate Mischung der Behaviors in der durch die Peephole-Perzepte $P_{\mathcal{B}_j}$ charakterisierten Situation. Zur Inferenz dieser Mischung wurde das *Behavior-Classification*-Modell $\mathcal{B}_j$ formuliert, das dafür nur die entsprechenden Peephole-Perzepte $P_{\mathcal{B}_j}$ benötigt. Die Struktur von Bayesschen Programmen (vergl. Bild 1) ermöglicht es, (bedingte) Wahrscheinlichkeitsverteilungen eines Bayesschen Programms als Fragen an andere Bayessche Programme zu definieren. In der funktionalen Programmierung würde man von Abstützung bzw. Ersetzung eines Ausdrucks durch einen Funktionsaufruf sprechen. Entsprechend kann die Verteilung $P\left(B_{Man_j} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right)$ als Frage (4) an das *Behavior-Classification*-Modell $\mathcal{B}_j$ umformuliert bzw. durch die Frage (4) ersetzt (7) werden:

$$(7) \quad P\left(B_{Man_j} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right) \equiv P\left(B_{Man_j} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right).$$

Eine solche Umformulierung bzw. Ersetzung erzeugt eine *Verbindung* (Bessière et al., 2008) zwischen dem *Gating*-Modell $\mathcal{G}_j$ und dem *Behavior-Classification*-Modell $\mathcal{B}_j$, bei dem das *Behavior-Classification*-Modell $\mathcal{B}_j$ als eine Art Unterprogramm des *Gating*-Modells $\mathcal{G}_j$ fungiert. Der letzte Term $P\left(A \mid B_{Man_j}, P_{\mathcal{A}_j}, \pi_{\mathcal{G}_j}\right)$ impliziert eine Wahrscheinlichkeitsverteilung $P\left(A \mid B_{Man_j} = beh_{j_k}, P_{\mathcal{A}_j}, \pi_{\mathcal{G}_j}\right), k = 1, \ldots n_j$ für jede mögliche Belegung der Behavior-Variablen B_{Man_j}. Die Vorhersage von Fahraktionen unter dem Wissen, dass das Behavior $B_{Man_j} = beh_{j_k}$ aktiv ist, kann durch das entsprechende *Action*-Modell $\mathcal{A}_{j_k}$ inferiert werden, das zur Inferenz jedoch nur die entsprechenden Peephole-Perzepte $P_{\mathcal{A}_{j_k}}$ benötigt. Folglich wird die Verteilung $P\left(A \mid B_{Man_j}, P_{\mathcal{A}_j}, \pi_{\mathcal{G}_j}\right)$ für jede mögliche Belegung $B_{Man_j} = beh_{j_k}, k = 1, \ldots, n_j$ als Frage an das *Action*-Modell $\mathcal{A}_{j_k}$ umformuliert (8), das das entsprechende Behavior $Behavior_{j_k}$ modelliert:

$$(8)\quad P\left(A \mid B_{Man_j} = beh_{j_k}, P_{\mathcal{A}_j}, \pi_{\mathcal{G}_j}\right) \equiv P\left(A \mid P_{\mathcal{A}_{j_k}}, \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right), \qquad k = 1, \dots, n_j.$$

Diese Abstützungen bzw. Ersetzungen definieren eine Verbindung zwischen dem *Gating*-Modell $\mathcal{G}_j$ und den n_j *Action*-Modellen $\mathcal{A}_{j_k}, k = 1, \dots, n_j$, die dadurch als Unterprogramme des *Gating*-Modells verwendet werden können. Durch die Anfragen an das *Behavior-Classification*-Modell und die *Action*-Modelle kann man die Dekomposition des *Gating*-Modells $\mathcal{G}_j$ (6) also weiter vereinfachen zu:

$$(9)\ P\left(A, B_{Man_j}, P_{\mathcal{A}_j}, P_{\mathcal{B}_j} \mid \pi_{\mathcal{G}_j}\right) =$$
$$P\left(P_{\mathcal{A}_j}, P_{\mathcal{B}_j} \mid \pi_{\mathcal{G}_j}\right) \cdot P\left(B_{Man_j} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right) \cdot P\left(A \mid P_{\mathcal{A}_{j_k}}, \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right), k = 1, \dots, n_j.$$

Verwendungszweck

Die Frage, die das *Gating*-Modell $\mathcal{G}_j$ zur Erzeugung von komplexen Verhalten beantworten soll, wird wie folgt definiert:

$$(10)\ P\left(A \mid P_{\mathcal{A}_j}, P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right).$$

Wie sich leicht mit dem *Gating*-Modell (6) zeigen lässt, entspricht diese Frage einer gewichten Summe über alle zu mischenden Behaviors:

$$(11)\ P\left(A \mid P_{\mathcal{A}_j}, P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right) = \sum_{B_{Man_j}} P\left(B_{Man_j} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right) \cdot P\left(A \mid B_{Man_j}, P_{\mathcal{A}_j}, \pi_{\mathcal{G}_j}\right).$$

Dank der Abstützungen (7) und (8) lassen sich noch einige Vereinfachungen vornehmen.

$$(12)\quad P\left(A \mid P_{\mathcal{A}_j}, P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right) = \sum_{B_{Man_j}} P\left(B_{Man_j} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right) \cdot P\left(A \mid P_{\mathcal{A}_{j_k}}, \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right)$$
$$= \sum_{k=1}^{n_j} \left[P\left(B_{Man_j} = beh_{j_k} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right) \cdot P\left(A \mid P_{\mathcal{A}_{j_k}}, \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right)\right].$$

Eine Zusammenfassung der Modelle findet sich in Bild 3.

Modell, Inferenz	**Zerlegung**	**Frage**
Action-(Classification-)Modell $\mathcal{A}_{j_k}$	(1) $P\left(A, P_{\mathcal{A}_{j_k}} \mid \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right)$ $= P\left(A \mid \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right) \cdot \prod_{l=1}^{p} P\left(P_{j_{k_l}} \mid A, \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right)$	(2) $P\left(A \mid P_{\mathcal{A}_{j_k}}, \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right)$

Behavior-Classification-Modell $\mathcal{B}_j$	(3) $P\left(B_{Man_j}, P_{\mathcal{B}_j} \mid \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right)$ $= P\left(B_{Man_j} \mid \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right)$ $\cdot \prod_{l=1}^{q} P\left(P_{j_l} \mid B_{Man_j}, \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right)$	(4) $P\left(B_{Man_j} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right)$
Gating-Modell $\mathcal{G}_j$	(6.1) $P\left(A, B_{Man_j}, P_{\mathcal{A}_j}, P_{\mathcal{B}_j} \mid \pi_{\mathcal{G}_j}\right)$ $= P\left(P_{\mathcal{A}_j}, P_{\mathcal{B}_j} \mid \pi_{\mathcal{G}_j}\right) \cdot P\left(B_{Man_j} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right)$ $\cdot P\left(A \mid B_{Man_j}, P_{\mathcal{A}_j}, \pi_{\mathcal{G}_j}\right)$ mit den Abstützungen bzw. Ersetzungen (7) $P\left(B_{Man_j} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right)$ $\equiv P\left(B_{Man_j} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right)$ (8) $P\left(A \mid B_{Man_j} = beh_{j_k}, P_{\mathcal{A}_j}, \pi_{\mathcal{G}_j}\right)$ $\equiv P\left(A \mid P_{\mathcal{A}_{j_k}}, \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right)$, $k = 1, \dots, n_j$. (6.2) $P\left(A, B_{Man_j}, P_{\mathcal{A}_j}, P_{\mathcal{B}_j} \mid \pi_{\mathcal{G}_j}\right)$ $= P\left(P_{\mathcal{A}_j}, P_{\mathcal{B}_j} \mid \pi_{\mathcal{G}_j}\right) \cdot P\left(B_{Man_j} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right)$ $\cdot P\left(A \mid P_{\mathcal{A}_{j_k}}, \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right)$, $k = 1, \dots, n_j$.	(10) $P\left(A \mid P_{\mathcal{A}_j}, P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right)$
Inferenz der Aktion A als *Mixture-of-Behaviors*	(12) $P\left(A \mid P_{\mathcal{A}_j}, P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right)$ $= \sum_{k=1}^{n_j} \left[P\left(B_{Man_j} = beh_{j_k} \mid P_{\mathcal{B}_j}, \pi_{\mathcal{B}_j}, \delta_{\mathcal{B}_j}\right) \cdot P\left(A \mid P_{\mathcal{A}_{j_k}}, \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right)\right]$	(10) $P\left(A \mid P_{\mathcal{A}_j}, P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right)$

Bild 3: Zusammenfassende Darstellung der Modelle und der Fragen, die durch die Modelle beantwortet werden können: das Gating-Modell $\mathcal{G}_j$ beantwortet Fragen $P\left(A \mid P_{\mathcal{A}_j}, P_{\mathcal{B}_j}, \pi_{\mathcal{G}_j}\right)$, indem es sich selbst wieder auf das Behavior-Classification-Modell $\mathcal{B}_j$ und auf die Action-(Classification-)le $\mathcal{A}_{j_k}, k = 1, \dots, n_j$ mit jeweils relevanten Perzepten $P_{\mathcal{B}_j}$ und $P_{\mathcal{A}_{j_k}}$ abstützt.

2.2.4 MoB Assemblages

Das *Gating*-Modell $\mathcal{G}_j$ kombiniert die n_j *Action*-Modelle $\mathcal{A}_{j_k}, k = 1, \ldots, n_j$ und das *Behavior-Classification*-Modell $\mathcal{B}_j$ zu einem Modellkonstrukt für die Erzeugung von komplexen Fahrverhalten für das $Maneuver_j$. Ein solches Modellkonstrukt bezeichnen wir in Anlehnung an Arkin (1998) als *Assemblage* (siehe Bild 4).

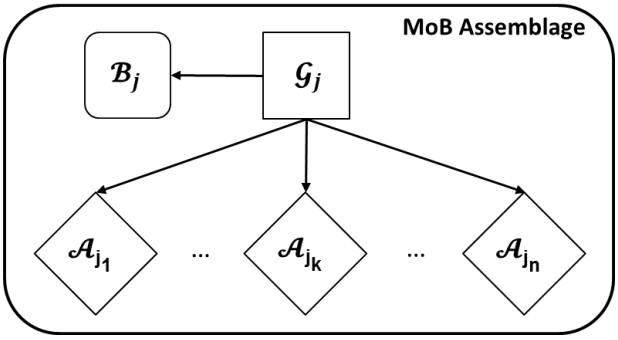

Bild 4: Grafische Repräsentation der Struktur eines modularen BAD MoB Modells für die in Bild 2 gezeigte Verhaltenshierarchie über das $Maneuver_j$. Quadratische Knoten repräsentieren Gating-, abgerundete Knoten repräsentieren Behavior-Classification- und karoförmige Knoten repräsentieren Action-(Classification-)Modelle (Notation angelehnt an Bishop und Svensen (2003)). Gerichtete Verbindungen besagen, dass eine bedingte Wahrscheinlichkeitsverteilung des Eltern-Modells als Frage an das Kind-Modell definiert ist.

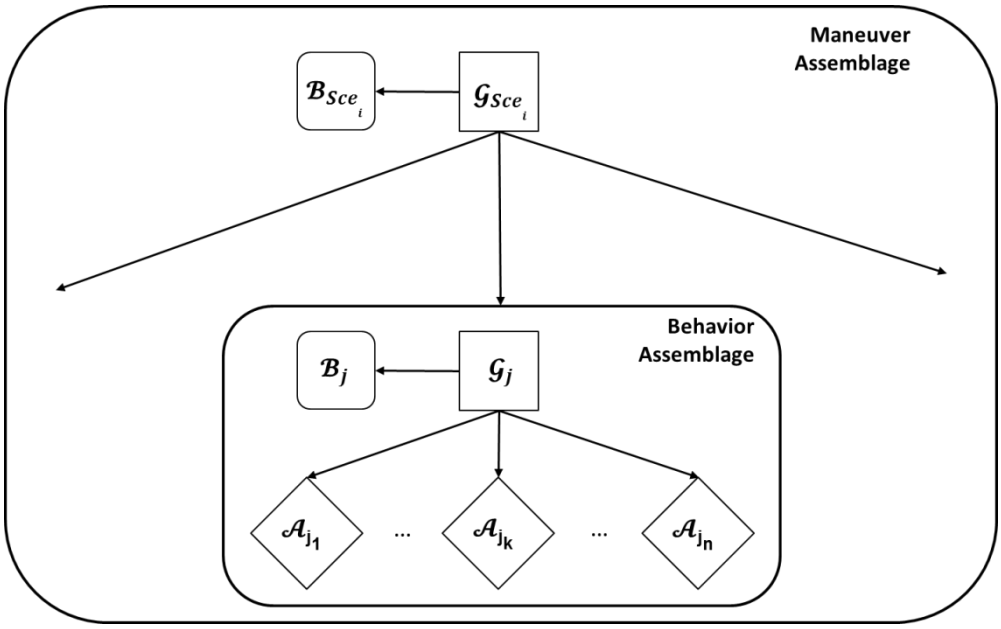

Bild 5: Grafische Repräsentation der Struktur eines modularen BAD MoB Modells für die in Bild 2 gezeigte Verhaltenshierarchie. Die MoB Assemblages für die jeweiligen $Maneuver_1$ bis $Maneuver_m$ wurden durch ein weiteres Behavior-Classification- und Gating-Modell zu einem modularen BAD MoB Modell zur Modellierung des $Scenario_i$ kombiniert.

MoB Assemblages lassen sich wie einfache *Action*-Modelle verwenden und in Verbindung mit einem weiteren *Behavior-Classification*- durch ein weiteres *Gating*-Modell zu einem hierarchischen MoB Assemblage kombinieren (siehe Bild 5). Dafür müssen in dem *Gating*-Modell (6.2) die Fragen an die *Action*-Modelle $P\left(A \mid P_{\mathcal{A}_{j_k}}, \pi_{\mathcal{A}_{j_k}}, \delta_{\mathcal{A}_{j_k}}\right)$ (2) einfach durch

Fragen an die *Gating*-Modelle (10) der zu kombinierenden MoB Assemblages ersetzt werden. Dabei muß die Perzeptmenge des oberen *Gating*-Modells in die zwei Perzeptmengen des unteren *Gating*-Modells zerlegt werden.

Als Beispiel sei ausschnittsweise eine solche Zuweisung für das in Bild 5 gezeigte *Gating*-Modell $\mathcal{G}_{Sce_i}$ gezeigt:

$$P\left(A \mid B_{Sce_i} = man_j, P_{\mathcal{G}}, \pi_{\mathcal{G}_{Sce_i}}\right) \equiv P\left(A \mid P_{\mathcal{G}_j}, \pi_{\mathcal{G}_j}\right),$$

$$\text{mit } P_{\mathcal{G}} = \{P_{\mathcal{G}_k}, k = 1, \dots, m\}, P_{G_j} = \{P_{\mathcal{A}_j}, P_{\mathcal{B}_j}\}, j = 1, \dots, m.$$

Das *Gating*-Modell $\mathcal{G}_{Sce_i}$ braucht dabei keine weitere Kenntnis über die spätere Verteilung der Peephole-Perzepte $P_{G_j} = \{P_{\mathcal{A}_j}, P_{\mathcal{B}_j}\}$ in den *Gating*-Modellen $\mathcal{G}_j, j = 1, \dots, m$. *Diese* hierarchische Kombination erlaubt die Entwicklung modularer BAD MoB Modelle für Verhaltensbibliotheken beliebiger Komplexität und beliebig viele hierarchische Ebenen.

3 Umsetzung

Für eine erste Implementierung eines modularen BAD MoB Modells haben wir ein komplexes Fahrszenario „Fahren auf einer schwach befahrenen Landstraße“ gewählt, bei dem der zu modellierende Fahrer verschiedene Strecken der Rennsimulation TORCS[3] befährt. Bei Annäherung an langsamere Fahrzeuge soll diesen in kurzen Abstand gefolgt werden, bis sich eine Möglichkeit zum Überholen bietet. In Anlehnung an Möbus und Eilers (2010) wurde dieses Szenario als ein (*Driving) Scenario* aufgefasst, dass sich in drei (*Driving) Maneuvers* verfeinern lässt: *LaneFollowing*, *CarFollowing* und *Overtaking*. *LaneFollowing* hat sich schon in einer früheren Arbeit (Möbus und Eilers, 2009a) als zu komplex für eine zufriedenstellende Modellierung durch ein BAD Modell *ohne* Mixture-of-Behaviors herausgestellt und wird daher weiter in drei *laneFollowing.Behaviors* für das Fahren auf einer geraden Strecke (*Straight*), das Durchfahren einer weiten Kurve (*Wide*) und das Durchfahren einer engen Kurve (*Sharp*) aufgeteilt.

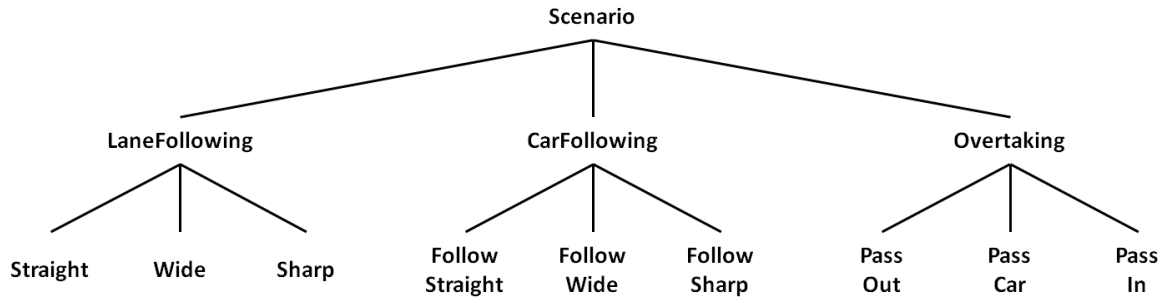

Bild 6: Verhaltenshierarchie für die Implementierung eines modularen BAD MoB Modells in TORCS.

Vergleichbar besteht *CarFollowing* aus den *carFollowing.Behaviors* für das Folgefahren auf einer geraden Strecke (*FollowStraight*), in einer weiten Kurve (*FollowWide*) und in einer scharfen Kurve (*FollowSharp*). Das dritte Manöver *Overtaking* ist aufgebaut aus den drei

[3] TORCS – The Open-Source Racing Simulation: http://torcs.sourceforge.net/ (besucht am 27.07.2010).

overtaking.Behaviors für das Ausscheren (*PassOut*), das Vorbeifahren (*PassCar*) und das Einscheren (*PassIn*). Die resultierende Verhaltenshierarchie ist in Bild 6 dargestellt.

Das modulare BAD MoB Modell für die in Bild 6 gezeigte Verhaltenshierarchie ist entsprechend aufgebaut aus vier *Gating-*, vier *Behavior-Classification-* und neun *Action*-Modellen, die insgesamt drei Kompetenzlevel abdecken (siehe Bild 7).

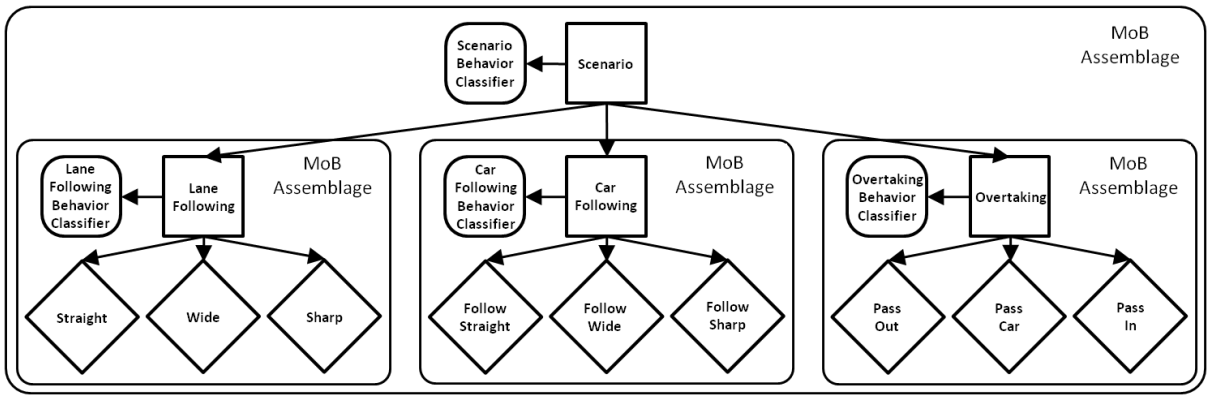

Bild 7: Hierarchische Struktur des modularen BAD MoB Modells in TORCS, aufgebaut aus vier Gating-, vier Behavior-Classification- und neun Action-Modellen.

3.1 Ergebnisse

Die ersten Ergebnisse sind sehr vielversprechend. Das implementierte BAD MoB Modell ist in der Lage, sowohl die zur Aufzeichnung der Lerndaten benutzten als auch neue, unbekannte Strecken von vergleichbarer Komplexität vollständig und weitgehend fehlerfrei zu befahren. Wie angestrebt, ist das BAD MoB Modell in der Lage, die komplexen Manöver für das Spurhalten (*LaneFollowing*), einfaches Folgefahren (*CarFollowing*) und Überholmanöver (*Overtaking*) durch das Mischen und Aneinanderreihen einfacherer Behaviors zu erzeugen. Ein Beispiel der Fähigkeiten des Modells ist in Bild 8 dargestellt (Videos sind verfügbar unter http://www.lks.uni-oldenburg.de/46350.html).

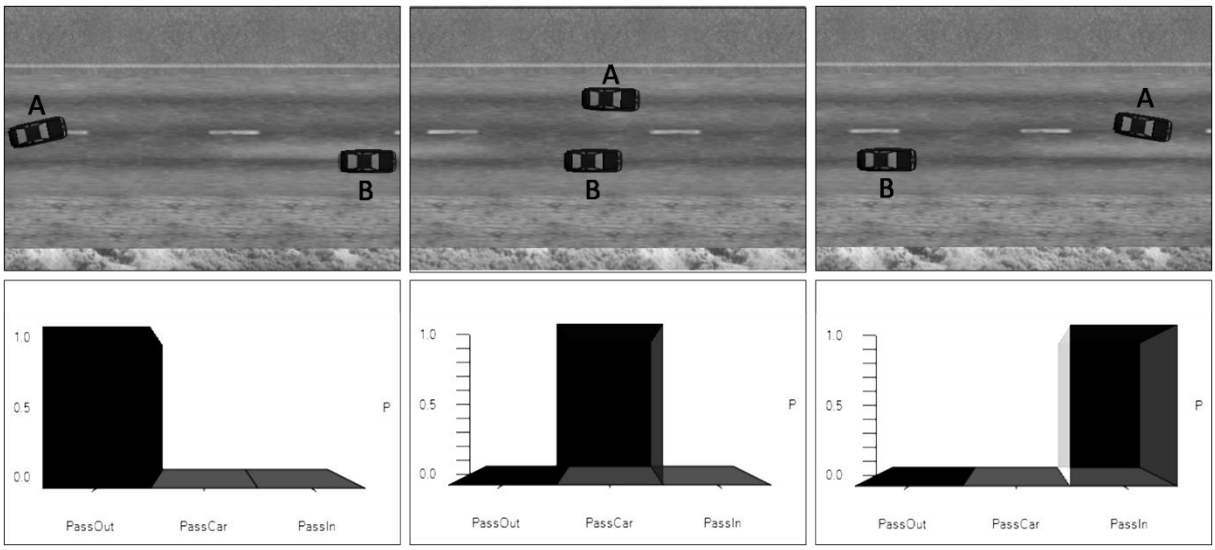

Bild 8: Sequenz von Verhalten während eines Überholmanövers. Die obere Reihe zeigt Screenshots des modularen BAD MoB Modells (A) beim Überholen eines langsameren Fahrzeugs (B) in der

TORCS Simulation. Die untere Reihe zeigt eine dazugehörige inferierte bedingte Verteilung über die einzelnen Behaviors des Überholmanövers.

4 Literatur

Anderson, J.R. (2002): Learning and Memory, John Wiley.

Arkin, R.C. (1998): Behavior-Based Robotics, MIT Press.

Bessière, P. and the BIBA INRIA Research Group (2003): Survey: Probabilistic Methodology and Techniques for Artefact Conception and Development, Technical Report RR-4730, INRIA.

Bessière, P., Laugier, Ch., and Siegwart, R. (eds.) (2008): Probabilistic Reasoning and Decision Making in Sensory-Motor Systems, Berlin: Springer, ISBN 978-3-540-79006-8.

Bishop, C. M. and Svensén, M. (2003): Bayesian hierarchical mixtures of experts. In: Kjaerulff, U. and C. Meek (Ed.): Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence, S. 57-64.

Eilers, M. and Möbus, C. (2010): Learning of a Bayesian Autonomous Driver Mixture-of-Behaviors (BAD-MoB) Model, in: Advances in Applied Digital Human Modeling, S. 436-445, Vincent G. Duffy (ed), CRC Press, Taylor & Francis Group, Boca Raton, 2010/2011, ISBN 978-1-4398-3511-1 and in: W. Karwowski and G. Salvendy (eds), 1st International Conference On Applied Digital Human Modeling, 17-20 July, 2010, Intercontinental, Miami Florida, USA, Conference Proceedings, Session Digital Human Modeling in the Bayesian Programming Framework, USA Publishing, ISBN-13: 978-0-9796435-4-5

Fitts, P.M. and Posner, M.I., Human Performance, Belmont, CA: Brooks/Cole, ISBN 0-13-445247-X, (1967).

Lebeltel, O., Bessière, P., Diard, J. and Mazer, E. (2004): Bayesian Robot Programming, Autonomous Robots 16, S. 49-79.

Möbus, C. and Eilers, M. (2008): First Steps Towards Driver Modeling according to the Bayesian Programming Approach, Symposium Cognitive Modeling, S.59, in L. Urbas, Th. Goschke and B. Velichkovsky (eds), KogWis2008. Christoph Hille, Dresden, ISBN 978-3-939025-14-6.

Möbus, C. and Eilers, M. (2009a): Further Steps Towards Driver Modeling according to the Bayesian Programming Approach, in: Vincent G. Duffy (Ed.), Digital Human Modeling, HCI 2009, San Diego, CA, USA, Springer: Lecture Notes in Computer Science (LNCS 5620) and Lecture Notes in Artificial Intelligence (LNAI), ISBN 978-3-642-02808-3, p. 413 - 422

Möbus, C., Eilers, M., Garbe, H., and Zilinski, M. (2009b): Probabilistic, and Empirical Grounded Modeling of Agents in Partial Cooperative (Traffic) Scenarios, in: Vincent G. Duffy (Ed.), Digital Human Modeling, Conference Proceedings, HCI 2009, San Diego, CA. USA, Springer: Lecture Notes in Computer Science (LNCS 5620) and Lecture Notes in Artificial Intelligence (LNAI), ISBN 978-3-642-02808-3, p. 423 - 432

Möbus, C., Eilers, M., Zilinski, M., and Garbe, H. (2009c): Mixture of Behaviors in a Bayesian Driver Model, in: Lichtenstein, Antje; Stößel, Christian; Clemens, Caroline (Hrsgb), Der Mensch im Mittelpunkt technischer Systeme, 8. Berliner Werkstatt, Mensch-Maschine-Systeme, 7.- 9. Oktober 2009, (ZMMS Spektrum Band 22), S.96 und S.221-226 (CD), Fortschr.-Ber., Mensch-Maschine-Systeme, VDI Reihe 22, Nr. 29, Düsseldorf: VDI Verlag 2009, ISBN 978-3-18-302922-8, ISSN 1439-958X

Möbus, C. and Eilers, M. (2010a): Mixture of Behaviors and Levels-of-Expertise in a Bayesian Autonomous Driver Model, in: Advances in Applied Digital Human Modeling, S. 425-435, Vincent G. Duffy (ed), CRC Press, Taylor & Francis Group, Boca Raton, 2010/2011, ISBN 978-1-4398-3511-1 and in: W. Karwowski and G. Salvendy (eds), 1st International Conference On Applied Digital Human Modeling, 17-20 July, 2010, Intercontinental, Miami Florida, USA, Conference Proceedings, Session Digital Human Modeling in the Bayesian Programming Framework, USA Publishing, ISBN-13: 978-0-9796435-4-5.

Möbus, C. and Eilers, M. (2010b): Integrating Anticipatory Competence into a Bayesian Driver Model, HMAT 2010 (Human Modeling in Assisted Transportation) Workshop, Heidelberg: Springer (in press)

Pearl, J.(2009): Causality – Models, Reasoning, and Inference, 2nd ed., Cambridge University Press, 2009, ISBN 978-0-521-89560-6.

Schwarz, G. (1978): Estimating the Dimension of a Model, The Annals of Statistics, Vol. 6, Nr. 2, S. 461-464.

5 Index

Bayesian Autonomous Driver Models, Mixture of Behaviors, Mixture of Experts, Bayesian Real-Time-Control, Levels of Expertise, Bayesian Programming