

Anhänge¹

A-1	Alphabetische Liste sämtlicher im Rahmen des MCIPEC spezifizierten offenen Atome und offenen Funktionsausdrücke.....	503
A-2	Anhang zur Simulationssprache Visual SLAM und zum Simulationssystem AweSim	625
A-2.1	Zulässige Operatoren auf der Ebene der graphischen Programmierung von Visual SLAM	625
A-2.2	Netzwerkbezogene Variablen in Visual SLAM	626
A-2.3	Grundlegende Netzwerkelemente in Visual SLAM	627
A-2.3.1	Blöcke	628
A-2.3.2	Knotenverbindungselemente.....	632
A-2.3.3	Knoten.....	635
A-2.3.4	Funktionen	662
A-2.4	Zu grundlegenden subnetzwerkbezogenen Sprachelementen in Visual SLAM	668
A-2.4.1	Variablen.....	668
A-2.4.2	Netzwerk- und Subnetzwerkelemente	669
A-2.4.3	Funktionen	676
A-2.5	Zu User Insert-spezifischen Sprachelementen in Visual SLAM in Verbindung mit der Programmiersprache C	676
A-2.5.1	Spezielle Netzwerk- und Subnetzwerkelemente.....	676
A-2.5.2	Benutzerspezifische Eingaben in Gestalt von C-Programmen	678
A-2.5.3	Spezielle Funktionen.....	679
A-2.6	Zu Steuerungsanweisungen in Visual SLAM.....	687
A-2.6.1	Steuerungsanweisungen in Steuerungsdateien.....	687
A-2.6.2	Die„LIMITSVSN“-Anweisung in objektorientierten Subnetzwerkklassem.....	704
A-3	Anhang zum Simulationsprojekt MCIPEC	705
A-3.1	Liste der im Simulationsprojekt MCIPEC enthaltenen Visual SLAM-Variablen (ohne Berücksichtigung von Feldvariablen)	705
A-3.2	Steuerungsanweisungen der Steuerungsdatei CVER_TP2	709
A-3.3	Steuerungsanweisungen der Steuerungsdatei CVER_TP6.....	710
A-3.4	AweSim SUMMARY-Report des Szenarios TP_VERI1.....	711

¹ Diese Anhänge gehören zu der von Markus Pütz verfassten Monographie mit dem Titel „Metacontrolling im partizipativen Entwicklungscontrolling – Ansätze zur formalsprachlichen Spezifikation und simulationsgestützten Modellierung“.

A-3.5	Daten der Ausgabedatei VEA _v l233.dat bezüglich des mit dem Szenario TP_VERI1 durchgeführten Simulationsexperimentes.....	712
A-3.6	AweSim SUMMARY-Report des Szenarios TP_VERI2.....	712
A-3.7	Daten der Ausgabedatei VEA _v l233.dat bezüglich des mit dem Szenario TP_VERI2 durchgeführten Simulationsexperimentes.....	713
A-3.8	Quellcode des benutzerspezifischen C-Programms UI_VEVA	714
A-3.9	Steuerungsanweisungen der Steuerungsdatei CVAL_TP7.....	717
A-3.10	AweSim SUMMARY-Report des Szenarios TP_VALI.....	719
A-3.11	Daten der Ausgabedatei VEA _v l233.dat bezüglich des mit dem Szenario TP_VALI durchgeführten Simulationsexperimentes	720
A-3.12	Daten der Ausgabedatei VAPTrueBppzl233.dat bezüglich des mit dem Szenario TP_VALI durchgeführten Simulationsexperimentes	720

A-1 Alphabetische Liste sämtlicher im Rahmen des MCIPEC spezifizierten offenen Atome und offenen Funktionsausdrücke

Offener Funktionsausdruck	Bedeutung	SNr.
aAnpassVar(j)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt die aktuelle Anpassungsvariante des Prozesses (j). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden Prozessen (j) jeweils zum Abschluss der originären bzw. aktualisierten Spezifikation des Prozesses (j) in Form einer natürlichen Zahl einschließlich der Zahl „0“ zugewiesen.	27
abedG(j,vpnl)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt das aktuelle bedingte Gewicht des Prozesses (j) unter Berücksichtigung seiner Vorgängerprozessnummernliste (vpnl), wobei von einem zyklensfreien offenen Entscheidungsnetz (OEN) ausgegangen wird. <u>Funktionsgleichung</u>: Bei der Funktionswertermittlung werden vier Fälle unterschieden². Nachfolgend sei dazu das lokale Gewicht von (j) mit g_j und das bedingte Gewicht mit g_j^* bezeichnet; ferner sei der k-te unmittelbare Vorgänger von (j) mit $v_k(j)$, dessen bedingtes Gewicht mit $g_{v_k(j)}^*$, das von $v_k(j)$ ausgehende lokale Gewicht von (j) mit g_{j,v_k}, ein beliebiger unmittelbarer Vorgänger von (j) mit $v_{\bar{k}}(j)$, und die Anzahl der unmittelbaren Vorgänger von (j) mit r bezeichnet; $g_j, g_{j,v_k}, g_{v_k(j)}^*, g_{v_{\bar{k}}(j)}^* \in]0, 1]$; $k \in \mathbb{N}_{>0}$, $r \in \mathbb{N}$. Fall 1): liegt vor, wenn (j) keinen unmittelbaren Vorgänger hat³; sodann gilt⁴: $\text{abedG}(j, \text{vpnl}) := 1$. Fall 2): liegt vor, wenn (j) genau einen unmittelbaren Vorgänger hat und der Prozessbeginnknoten von (j) eine konjunkte Eingangsseite aufweist,	42

² Vgl. dazu die Hinweise zur Ermittlung von g_{ie}^* in Gleichung (1) in Abschnitt 2.2.1.2.1.

³ In diesem Fall enthält die Vorgängerprozessnummerliste (vpnl) des aktuellen Prozesses (j), nachfolgend durch die Individuenkonstante (Aktj) repräsentiert, die „unechte“ Prozessnummer „None“, was folgende wff zum Ausdruck bringt: $(\exists \text{vpnl}) \text{Prozess_VPNListe}(\text{Aktj}, \text{vpnl}) \wedge \text{VPNListe_VProzNr}(\text{vpnl}, \text{None})$.

⁴ Dies kennzeichnet folgende wff: $(\exists \text{vpnl}) \text{Prozess_VPNListe}(\text{Aktj}, \text{vpnl}) \wedge \neg \text{VPNListe_VProzNr}(\text{vpnl}, \text{None}) \wedge \text{Prozess_AKEKennz}(\text{Aktj}, \text{EKon})$.

⁵ Dies kennzeichnet folgende wff: $\exists (\text{vpnl}) \text{Prozess_VPNListe}(\text{Aktj}, \text{vpnl}) \wedge \neg \text{VPNListe_VProzNr}(\text{vpnl}, \text{None}) \wedge \text{Prozess_AKEKennz}(\text{Aktj}, \text{EEx})$.

	dann gilt: $\text{abedG}(j, \text{vpnl}) := g_{v_1(j)}^* \cdot g_{j, v_1}$. Fall 3): liegt vor, wenn (j) mindestens zwei unmittelbare Vorgänger hat und der Prozessbeginnknotten von (j) eine konjunkte Eingangsseite aufweist, dann gilt: $\text{abedG}(j, \text{vpnl}) := g_{v_k(j)}^* \cdot g_{j, v_k}$. Fall 4): liegt vor, wenn (j) mehrere unmittelbare Vorgänger hat und der Prozessbeginnknotten von (j) eine exklusiv disjunkte Eingangsseite besitzt⁵; dann gilt: $\text{abedG}(j, \text{vpnl}) := \sum_{k=1}^r g_{v_k(j)}^* \cdot g_{j, v_k}$.	
agBt_j(j, agdl, agd, k=5)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Arbeitsgangdatenvektorenliste (agdl) enthaltenen Arbeitsgangdatenvektors (agd) zu dessen Komponente k=5; dieser Wert repräsentiert die Bearbeitungszeit des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt über Annahmen für die relevanten Prozesse in Form einer nicht negativen reellen Zahl zugeordnet.	73
agBZK_j(j, agdl, agd, k=6)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Arbeitsgangdatenvektorenliste (agdl) enthaltenen Arbeitsgangdatenvektors (agd) zu dessen Komponente k=6; der Wert repräsentiert das Bearbeitungszeitkennzeichen des betreffenden Arbeitsgangs. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt über Annahmen für die relevanten Prozesse in Form der Werte „1“ zur Kennzeichnung einer Stückbearbeitungszeit und „2“ zur Kennzeichnung einer Chargenzeit zugeordnet.	74

Offenes Atom	Bedeutung	SNr.
AgDatV(agd)	Etwas (agd) ist ein Arbeitsgangdatenvektor.	64
AgDatV_Komp(agd,k)	Ein Arbeitsgangdatenvektor (agd) weist die Komponente (k) auf.	67
AgDatVL_AgDatV(agdl, agd)	Eine Arbeitsgangdatenvektorenliste (agdl) beinhaltet einen Arbeitsgangdatenvektor (agd).	66
AgDatVListe(agdl)	Etwas (agdl) ist eine Liste arbeitsgangspezifischer Datenvektoren (kurz Arbeitsgangdatenvektorenliste).	63
AgIAKL_AgIPBV(agial, agia₂)	Eine Liste (agial) realisierter arbeitsgangspezifischer Aktionskomponenten beinhaltet den realisierten arbeitsgangspezifischen Potenzialbelegungsvektor (agia ₂).	130
AgIAKL_AgIPVV(agial, agia₁)	Eine Liste (agial) realisierter arbeitsgangspezifischer Aktionskomponenten beinhaltet den realisierten arbeitsgangspezifischen Potenzialverbrauchsvektor (agia ₁).	128
AgIAKL_IAGATP(agial, agia₃)	Eine Liste (agial) realisierter arbeitsgangspezifischer Aktionskomponenten beinhaltet die realisierte arbeitsgangspezifische Anfangsterminposition (agia ₃).	132
AgIAKL_IAGETP(agial, agia₄)	Eine Liste (agial) realisierter arbeitsgangspezifischer Aktionskomponenten beinhaltet die realisierte arbeitsgangspezifische Endterminposition (agia ₄).	134
AgIstAKListe(agial)	Etwas (agial) ist eine Liste realisierter arbeitsgangspezifischer Aktionskomponenten.	125
AgIstPBelegV(agia₂)	Etwas (agia ₂) ist ein realisierter arbeitsgangspezifischer Potenzialbelegungsvektor ⁶ .	129
AgIstPVV(agia₁)	Etwas (agia ₁) ist ein realisierter arbeitsgangspezifischer Potenzialverbrauchsvektor ⁷ .	127
AgIstRV(agir)	Etwas (agir) ist ein realisierter arbeitsgangspezifischer Reaktionsgrößenvektor.	206
AgIstRVL_AgIstRV(agirl, agir)	Eine Liste (agirl) realisierter arbeitsgangspezifischer Reaktionsgrößenvektoren beinhaltet einen realisierten arbeitsgangspezifischen Reaktionsgrößenvektor (agir).	207
AgIstRV_Komp(agir,k)	Ein realisierter arbeitsgangspezifischer Reaktionsgrößenvektor (agir) weist die Komponente (k) auf.	208

⁶ AgIstPBelegV(agia₂) repräsentiert den realisierten Potenzialbelegungsvektor *agia_{j,2}* innerhalb der Spezifikation des Parameters *ap₅(up)*.

⁷ AgIstPVV(agia₁) repräsentiert den realisierten Potenzialverbrauchsvektor *agia_{j,1}* innerhalb der Spezifikation des Parameters *ap₅(up)*.

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
AgIstRVListe(agirl)	Etwas (agirl) ist eine Liste realisierter arbeitsgangspezifischer Reaktionsgrößenvektoren.	205
agNr_j(j,agdl,agd,k=2)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Arbeitsgangdatenvektorenliste (agdl) enthaltenen Arbeitsgangdatenvektors (agd) zu dessen Komponente k=2; dieser Wert repräsentiert die Arbeitsgangnummer des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden direkt über Annahmen für die relevanten Prozesse in Form einer natürlichen Zahl zugeordnet.	70
agPos_j(j,agdl,agd,k=1)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Arbeitsgangdatenvektorenliste (agdl) enthaltenen Arbeitsgangdatenvektors (agd) zu dessen Komponente k=1 (d.h. der ersten Komponente); dieser Wert repräsentiert die Arbeitsgangposition des betreffenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden direkt über Annahmen für die relevanten Prozesse in Form einer natürlichen Zahl zugeordnet.	69
agPSNr_j(j,agdl,agd,k=3)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Arbeitsgangdatenvektorenliste (agdl) enthaltenen Arbeitsgangdatenvektors (agd) zu dessen Komponente k=3; dieser Wert repräsentiert die Produktionsstufennummer des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden direkt über Annahmen für die relevanten Prozesse in Form einer natürlichen Zahl zugeordnet.	71
AgPTermV(agp)	Etwas (agp) ist ein Arbeitsgangplanterminvektor.	101
AgPTermV_Komp(agp,k)	Ein Arbeitsgangplanterminvektor (agp) weist die Komponente (k) auf.	103
AgPTermVListe(agpt)	Etwas (agpt) ist eine Liste arbeitsgangspezifischer Planterminvektoren (kurz Arbeitsgangplanterminvektorenliste).	100
AgPTVL_AgPTermV(agpt, agp)	Eine Arbeitsgangplanterminvektorenliste (agpt) beinhaltet den Arbeitsgangplanterminvektor (agp).	102

Offener Funktionsausdruck	Bedeutung	SNr.
agPAT_j(j,agpt,agp,k=1)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Arbeitsgangplanterminvektorenliste (agpt) enthaltenen Arbeitsgangvektors (ag) zu dessen Komponente k=1 (d.h. der ersten Komponente); dieser Wert repräsentiert den gegebenenfalls vorliegenden Plananfangstermin des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der für die relevanten Prozesse gegebenenfalls vorliegenden Planungsinformationen festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch kein Plananfangstermin für den hier relevanten Arbeitsgang (agp) gegeben ist; in diesem Fall gilt: $\text{agPAT}_j(j, \text{agpt}, \text{agp}, k=1) := -1.$ Fall 2): liegt vor, wenn ein Plananfangstermin für den hier relevanten Arbeitsgang (agp) gegeben ist; sodann gilt: $\text{agPAT}_j(j, \text{agpt}, \text{agp}, k=1) := t, \quad t \in \mathbb{R}_+.$	105
agPET_j(j,agpt,agp,k=2)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Arbeitsgangplanterminvektorenliste (agpt) enthaltenen Arbeitsgangvektors (ag) zu dessen Komponente k=2; dieser Wert repräsentiert den gegebenenfalls vorliegenden Planendtermin des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der für die relevanten Prozesse gegebenenfalls vorliegenden Planungsinformationen festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch kein Planendtermin für den hier relevanten Arbeitsgang (agp) gegeben ist; in diesem Fall gilt: $\text{agPET}_j(j, \text{agpt}, \text{agp}, k=1) := -1.$ Fall 2): liegt vor, wenn ein Plananfangstermin für den hier relevanten Arbeitsgang (agp) gegeben ist; sodann gilt: $\text{agPET}_j(j, \text{agpt}, \text{agp}, k=1) := t, \quad \text{mit } t \in \mathbb{R}_+ \text{ und}$ $\text{agPET}_j(j, \text{agpt}, \text{agp}, k=1) \geq \text{agPAT}_j(j, \text{agpt}, \text{agp}, k=1).$	106

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
agRt_j(j,agdl,agd,k=4)	<u>Funktionsbeschreibung</u> : Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Arbeitsgangdatenvektorenliste (agdl) enthaltenen Arbeitsgangdatenvektors (agd) zu dessen Komponente k=4; dieser Wert repräsentiert die Rüstzeit des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung</u> : Die einzelnen Funktionswerte werden direkt über Annahmen für die relevanten Prozesse in Form einer nicht negativen reellen Zahl zugeordnet.	72
AgSAKL_AgSPBV(agsal, agsa₂)	Eine Liste simulierter arbeitgangspezifischer Aktionskomponenten (agsal) beinhaltet den simulierten arbeitgangspezifischen Potenzialbelegungsvektor (agsa ₂).	114
AgSAKL_AgSPVV(agsal, agsa₁)	Eine Liste simulierter arbeitgangspezifischer Aktionskomponenten (agsal) beinhaltet den simulierten arbeitgangspezifischen Potenzialverbrauchsvektor (agsa ₁).	112
AgSAKL_SAgATP(agsal, agsa₃)	Etwas (agsal) beinhaltet (agsa ₃).	116
AgSAKL_SAgETP(agsal, agsa₄)	Etwas (agsal) beinhaltet (agsa ₄).	118
AgSimAKListe(agsal)	Etwas (agsal) ist eine Liste simulierter arbeitgangspezifischer Aktionskomponenten.	109
AgSimPBelegV(agsa₂)	Etwas (agsa ₂) ist ein simulierter arbeitgangspezifischer Potenzialbelegungsvektor ⁸ .	113
AgSimPVV(agsa₁)	Etwas (agsa ₁) ist ein simulierter arbeitgangspezifischer Potenzialverbrauchsvektor ⁹ .	111
AgSimRV(agsr)	Etwas (agsr) ist ein simulierter arbeitgangspezifischer Reaktionsgrößenvektor.	141
AgSimRV_Komp(agsr,k)	Ein simulierter arbeitgangspezifischer Reaktionsgrößenvektor (agsr) weist die Komponente (k) auf.	143
AgSimRVL_AgSimRV(agsrl, agsr)	Eine Liste simulierter arbeitgangspezifischer Reaktionsgrößenvektoren (agsrl) beinhaltet einen simulierten arbeitgangspezifischen Reaktionsgrößenvektor (agsr).	142
AgSimRVListe(agsrl)	Etwas (agsrl) ist eine Liste simulierter arbeitgangspezifischer Reaktionsgrößenvektoren.	140

⁸ AgSimPBelegV(agsa₂) repräsentiert den Potenzialbelegungsvektor $agsa_{j,2}$ innerhalb der Spezifikation des Parameters $ap_2(up)$.

⁹ AgSimPVV(agsa₁) repräsentiert den Potenzialverbrauchsvektor $agsa_{j,1}$ innerhalb der Spezifikation des Parameters $ap_2(up)$.

Offener Funktionsausdruck	Bedeutung	SNr.
agSoIstATA_j(j,agsil,agsi, k=1,agpt,agial)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste arbeitsgangspezifischer Soll-Ist-Abweichungsgrößenvektoren (agsil) enthaltenen arbeitsgangspezifischen Soll-Ist-Abweichungsgrößenvektors (agsi) zu dessen Komponente k=1 (d.h. der ersten Komponente) unter Berücksichtigung der Arbeitsplanterminvektorenliste (agpt) des betreffenden Prozesses sowie seiner Liste realisierter arbeitsgangspezifischer Aktionskomponenten (agial); dieser Wert repräsentiert die Soll-Ist-Anfangsterminabweichung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 105, 137, 219 und 220 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $\text{agPAT}_j(j, \text{agpt}, \text{kIKon}_{\text{agsi_agp}}(j, \text{agsil}, \text{agsi}, \text{agpt}), k=1)) = -1$ (d.h., es liegt noch kein Plananfangstermin für den hier relevanten prozessspezifischen Arbeitsgang vor) oder wenn gilt: $\text{iAgAT}_j(j, \text{agial}, \text{kIKon}_{\text{agsi_agia3}}(j, \text{agsil}, \text{agsi}, \text{agial})) = -1$ (d.h., es liegt noch kein realisierter Anfangstermin für den hier relevanten Arbeitsgang vor), dann folgt daraus: $\text{agSoIstATA}_j(j, \text{agsil}, \text{agsi}, k=1, \text{agpt}, \text{agial}) := 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $\text{agSoIstATA}_j(j, \text{agsil}, \text{agsi}, k=1, \text{agpt}, \text{agial}) :=$ $\text{agPAT}_j(j, \text{agpt}, \text{kIKon}_{\text{agsi_agp}}(j, \text{agsil}, \text{agsi}, \text{agpt}), k=1) -$ $\text{iAgAT}_j(j, \text{agial}, \text{kIKon}_{\text{agsi_agia3}}(j, \text{agsil}, \text{agsi}, \text{agial}))$.	221

Offener Funktionsausdruck	Bedeutung	SNr.
agSoIstETA_j(j,agsil,agsi,k=2,agpt,agial)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste arbeitgangspezifischer Soll-Ist-Abweichungsgrößenvektoren (agsil) enthaltenen arbeitgangspezifischen Soll-Ist-Abweichungsgrößenvektors (agsi) zu dessen Komponente k=2 unter Berücksichtigung der Arbeitsplanterminvektorenliste (agpt) des betreffenden Prozesses sowie seiner Liste realisierter arbeitgangspezifischer Aktionskomponenten (agial); dieser Wert repräsentiert die Soll-Ist-Endterminabweichung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 106, 138, 219 und 222 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $\text{agPET}_j(j, \text{agpt}, \text{kIKon}_{\text{agsi_agp}}(j, \text{agsil}, \text{agsi}, \text{agpt}), k=2) = -1$ (d.h., es liegt noch kein Planendtermin für den hier relevanten prozessspezifischen Arbeitsgang vor) oder wenn gilt: $\text{iAgET}_j(j, \text{agial}, \text{kIKon}_{\text{agsi_agia4}}(j, \text{agsil}, \text{agsi}, \text{agial})) = -1$ (d.h., es liegt noch kein realisierter Endtermin für den hier relevanten Arbeitsgang vor), dann folgt daraus: $\text{agSoIstETA}_j(j, \text{agsil}, \text{agsi}, k=2, \text{agpt}, \text{agial}) := 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor; folgt: $\text{agSoIstETA}_j(j, \text{agsil}, \text{agsi}, k=2, \text{agpt}, \text{agial}) :=$ $\text{agPET}_j(j, \text{agpt}, \text{kIKon}_{\text{agsi_agp}}(j, \text{agsil}, \text{agsi}, \text{agpt}), k=2) -$ $\text{iAgET}_j(j, \text{agial}, \text{kIKon}_{\text{agsi_agsa4}}(j, \text{agsil}, \text{agsi}, \text{agial}))$.	223

Offener Funktionsausdruck	Bedeutung	SNr.
agSoIstKA_j(j,agsil,agsi,k=4,agdl,agirl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste arbeitgangspezifischer Soll-Ist-Abweichungsgrößenvektoren (agsil) enthaltenen arbeitgangspezifischen Soll-Ist-Abweichungsgrößenvektors (agsi) zu dessen Komponente k=4 unter Berücksichtigung der Arbeitsgangdatenvektorenliste (agdl) des betreffenden Prozesses sowie seiner Liste realisierter arbeitgangspezifischer Reaktionsgrößenvektoren (agirl); dieser Wert repräsentiert die Soll-Ist-Kostenabweichung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 76, 212, 224 und 225 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $iAgKosten_j(j,agirl,kIKon_{agsi_agir}(j,agsil,agsi,agirl),k=3) = -1$ (d.h., es liegen noch keine Realisationsdaten über die Kosten des hier relevanten prozessspezifischen Arbeitsgangs vor), dann folgt daraus: $agSoIstKA_j(j,agsil,agsi,k=4,agdl,agirl) = 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $agSoIstKA_j(j,agsil,agss,k=4,agdl,agirl) :=$ $pAgKosten_j(j,agdl,kIKon_{agsi_agd}(j,agsil,agsi,agdl),k=8) -$ $iAgKosten_j(j,agirl,kIKon_{agsi_agir}(j,agsil,agsi,agirl),k=3)$.	227

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
agSoIstOpA_j(j,agsil,agsi,k=3,agdl,agirl)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste arbeitgangspezifischer Soll-Ist-Abweichungsgrößenvektoren (agsil) enthaltenen arbeitgangspezifischen Soll-Ist-Abweichungsgrößenvektors (agsi) zu dessen Komponente k=3 unter Berücksichtigung der Arbeitsgangdatenvektorliste (agpt) des betreffenden Prozesses sowie seiner Liste realisierter arbeitgangspezifischer Reaktionsgrößenvektoren (agirl); dieser Wert repräsentiert die Soll-Ist-Outputabweichung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 75, 210, 224 und 225 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $iAgOutput_j(j,agirl,kIKon_{agsi_agir}(j,agsil,agsi,agirl),k=1) = -1$ (d.h., es liegen noch keine Realisationsdaten über den Output des hier relevanten prozessspezifischen Arbeitsgangs vor), dann folgt daraus: $agSoIstOpA_j(j,agsil,agsi,k=3,agdl,agirl) = 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $agSoIstOpA_j(j,agsil,agsi,k=3,agdl,agirl) = pAgOutput_j(j,agdl,kIKon_{agsi_agd}(j,agsil,agsi,agdl),k=7) - iAgOutput_j(j,agirl,kIKon_{agsi_agir}(j,agsil,agsi,agirl),k=1)$.	226
AgSoIstV(agsi)	Etwas (agsi) ist ein arbeitgangspezifischer Soll-Ist-Abweichungsgrößenvektor.	215
AgSoIstV_Komp(agsi,k)	Ein arbeitgangspezifischer Soll-Ist-Abweichungsgrößenvektor (agsi) weist die Komponente (k) auf.	217
AgSoIstVL_AgSoIstV(agsil,agsi)	Eine Liste (agsil) simulierter arbeitgangspezifischer Reaktionsgrößenvektoren beinhaltet einen arbeitgangspezifischen Soll-Ist-Abweichungsgrößenvektor (agsi).	216
AgSoIstVListe(agsil)	Etwas (agsil) ist eine Liste arbeitgangspezifischer Soll-Ist-Abweichungsgrößenvektoren.	214

Offener Funktionsausdruck	Bedeutung	SNr.
agSoIstWsA_j(j,agsil,agsi,k=5,agdl,agirl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste arbeitgangspezifischer Soll-Ist-Abweichungsgrößenvektoren (agsil) enthaltenen arbeitgangspezifischen Soll-Ist-Abweichungsgrößenvektors (agsi) zu dessen Komponente k=5 unter Berücksichtigung der Arbeitsgangdatenvektorenliste (agdl) des betreffenden Prozesses sowie seiner Liste realisierter arbeitgangspezifischer Reaktionsgrößenvektoren (agirl); dieser Wert repräsentiert die Soll-Ist-Wertschöpfungsabweichung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 77, 213, 224 und 225 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $iAgWSchöpf_j(j,agirl,kIKon_{agsi_agir}(j,agsil,agsi,agirl),k=4) = -1$ (d.h., es liegen noch keine Simulationsdaten über die Wertschöpfung des hier relevanten prozessspezifischen Arbeitsgangs vor), dann folgt daraus: $agSoIstWsA_j(j,agsil,agsi,k=5,agdl,agirl) = 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $agSoIstWsA_j(j,agsil,agsi,k=5,agdl,agirl) =$ $pAgWSchöpf_j(j,agdl,kIKon_{agsi_agd}(j,agsil,agsi,agdl),k=9) -$ $iAgWSchöpf_j(j,agirl,kIKon_{agsi_agir}(j,agsil,agsi,agirl),k=4)$.	228

Offener Funktionsausdruck	Bedeutung	SNr.
agSoWiATA_j(j,agswl,agsw, k=1,agpt,agsal)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste arbeitgangspezifischer Soll-Wird-Abweichungsgrößenvektoren (agswl) enthaltenen arbeitgangspezifischen Soll-Wird-Abweichungsgrößenvektors (agsw) zu dessen Komponente k=1 (d.h. der ersten Komponente) unter Berücksichtigung der Arbeitsplanterminvektorenliste (agpt) des betreffenden Prozesses sowie seiner Liste simulierter arbeitgangspezifischer Aktionskomponenten (agsal); dieser Wert repräsentiert die Soll-Wird-Anfangsterminabweichung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 105, 121, 154 und 155 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $\text{agPAT}_j(j, \text{agpt}, \text{kIKon}_{\text{agsw_agp}}(j, \text{agswl}, \text{agsw}, \text{agpt}), k=1) = -1$ (d.h., es liegt noch kein Plananfangstermin für den hier relevanten prozessspezifischen Arbeitsgang vor) oder wenn gilt: $\text{sAgAT}_j(j, \text{agsal}, \text{kIKon}_{\text{agsw_agsa3}}(j, \text{agswl}, \text{agsw}, \text{agsal})) = -1$ (d.h., es liegt noch kein simulierter Anfangstermin für den hier relevanten Arbeitsgang vor), dann folgt daraus: $\text{agSoWiATA}_j(j, \text{agswl}, \text{agsw}, k=1, \text{agpt}, \text{agsal}) := 0.$ Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $\text{agSoWiATA}_j(j, \text{agswl}, \text{agsw}, k=1, \text{agpt}, \text{agsal}) :=$ $\text{agPAT}_j(j, \text{agpt}, \text{kIKon}_{\text{agsw_agp}}(j, \text{agswl}, \text{agsw}, \text{agpt}), k=1) -$ $\text{sAgAT}_j(j, \text{agsal}, \text{kIKon}_{\text{agsw_agsa3}}(j, \text{agswl}, \text{agsw}, \text{agsal})).$	156

Offener Funktionsausdruck	Bedeutung	SNr.
agSoWiETA_j(j,agswl,agsw, k=2,agpt,agsal)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste arbeitsgangspezifischer Soll-Wird-Abweichungsgrößenvektoren (agswl) enthaltenen arbeitsgangspezifischen Soll-Wird-Abweichungsgrößenvektors (agsw) zu dessen Komponente k=2 unter Berücksichtigung der Arbeitsplan-terminvektorenliste (agpt) des betreffenden Prozesses sowie seiner Liste simulierter arbeitsgangspezifischer Aktionskomponenten (agsal); dieser Wert repräsentiert die Soll-Wird-Endterminabweichung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 106, 122, 154 und 157 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $\text{agPET}_j(j, \text{agpt}, \text{kIKon}_{\text{agsw_agp}}(j, \text{agswl}, \text{agsw}, \text{agpt}), k=2) = -1$ (d.h., es liegt noch kein Planendtermin für den hier relevanten prozessspezifischen Arbeitsgang vor) oder wenn gilt: $\text{sAgET}_j(j, \text{agsal}, \text{kIKon}_{\text{agsw_agsa4}}(j, \text{agswl}, \text{agsw}, \text{agsal})) = -1$ (d.h., es liegt noch kein simulierter Endtermin für den hier relevanten Arbeitsgang vor), dann folgt daraus: $\text{agSoWiETA}_j(j, \text{agswl}, \text{agsw}, k=2, \text{agpt}, \text{agsal}) = 0.$ Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $\text{agSoWiETA}_j(j, \text{agswl}, \text{agsw}, k=2, \text{agpt}, \text{agsal}) = \text{agPET}_j(j, \text{agpt}, \text{kIKon}_{\text{agsw_agp}}(j, \text{agswl}, \text{agsw}, \text{agpt}), k=2) - \text{sAgET}_j(j, \text{agsal}, \text{kIKon}_{\text{agsw_agsa4}}(j, \text{agswl}, \text{agsw}, \text{agsal})).$	158

Offener Funktionsausdruck	Bedeutung	SNr.
agSoWiKA_j(j,agswl,agsw, k=4,agdl,agsrl)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste arbeitgangspezifischer Soll-Wird-Abweichungsgrößenvektoren (agswl) enthaltenen arbeitgangspezifischen Soll-Wird-Abweichungsgrößenvektors (agsw) zu dessen Komponente k=4 unter Berücksichtigung der Arbeitsgangdatenvektorenliste (agdl) des betreffenden Prozesses sowie seiner Liste simulierter arbeitgangspezifischer Reaktionsgrößenvektoren (agsrl); dieser Wert repräsentiert die Soll-Wird-Kostenabweichung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 76, 147, 159 und 160 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $sAgKosten_j(j,agsrl,kIKon_{agsw_agsr}(j,agswl,agsw,agsrl),k=3) = -1$ (d.h., es liegen noch keine Simulationsdaten über die Kosten des hier relevanten prozessspezifischen Arbeitsgangs vor), dann folgt daraus: $agSoWiKA_j(j,agswl,agsw,k=4,agdl,agsrl) = 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $agSoWiKA_j(j,agswl,agsw,k=4,agdl,agsrl) = pAgKosten_j(j,agdl,kIKon_{agsw_agd}(j,agswl,agsw,agdl),k=8) - sAgKosten_j(j,agsrl,kIKon_{agsw_agsr}(j,agswl,agsw,agsrl),k=3)$.	162

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
agSoWiOpA_j(j,agswl,agsw,k=3,agdl,agsrl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste arbeitgangspezifischer Soll-Wird-Abweichungsgrößenvektoren (agswl) enthaltenen arbeitgangspezifischen Soll-Wird-Abweichungsgrößenvektors (agsw) zu dessen Komponente k=3 unter Berücksichtigung der Arbeitsgangdatenvektorenliste (agdl) des betreffenden Prozesses sowie seiner Liste simulierter arbeitgangspezifischer Reaktionsgrößenvektoren (agsrl); dieser Wert repräsentiert die Soll-Wird-Outputabweichung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 75, 145, 159 und 160 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $sAgOutput_j(j,agsrl,kIKon_{agsw_agsr}(j,agswl,agsw,agsrl),k=1) = -1$ (d.h., es liegen noch keine Simulationsdaten über den Output des hier relevanten prozessspezifischen Arbeitsgangs vor), dann folgt daraus: $agSoWiOpA_j(j,agswl,agsw,k=3,agdl,agsrl) = 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $agSoWiOpA_j(j,agswl,agsw,k=3,agdl,agsrl) = pAgOutput_j(j,agdl,kIKon_{agsw_agd}(j,agswl,agsw,agdl),k=7) - sAgOutput_j(j,agsrl,kIKon_{agsw_agsr}(j,agswl,agsw,agsrl),k=1)$.	161
AgSoWiV(agsw)	Etwas (agsw) ist ein arbeitgangspezifischer Soll-Wird-Abweichungsgrößenvektor.	150
AgSoWiV_Komp(agsw,k)	Ein arbeitgangspezifischer Soll-Wird-Abweichungsgrößenvektor (agsw) weist die Komponente (k) auf.	152
AgSoWiVL_AgSoWiV(agswl,agsw)	Eine Liste (agswl) arbeitgangspezifischer Soll-Wird-Abweichungsgrößenvektoren beinhaltet den arbeitgangspezifischen Soll-Wird-Abweichungsgrößenvektor (agsw).	151
AgSoWiVListe(agswl)	Etwas (agswl) ist eine Liste arbeitgangspezifischer Soll-Wird-Abweichungsgrößenvektoren.	149

Offener Funktionsausdruck	Bedeutung	SNr.
agSoWiWsA_j(j,agswl,agsw,k=5,agdl,agsrl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste arbeitgangspezifischer Soll-Wird-Abweichungsgrößenvektoren (agswl) enthaltenen arbeitgangspezifischen Soll-Wird-Abweichungsgrößenvektors (agsw) zu dessen Komponente k=5 unter Berücksichtigung der Arbeitsgangdatenvektorenliste (agdl) des betreffenden Prozesses sowie seiner Liste simulierter arbeitgangspezifischer Reaktionsgrößenvektoren (agsrl); dieser Wert repräsentiert die Soll-Wird-Wertschöpfungsabweichung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 77, 148, 159 und 160 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $sAgWSchöpf_j(j,agsrl,kIKon_{agsw_agsr}(j,agswl,agsw,agsrl),k=5) = -1$ (d.h., es liegen noch keine Simulationsdaten über die Wertschöpfung des hier relevanten prozessspezifischen Arbeitsgangs vor), dann folgt daraus: $agSoWiWsA_j(j,agswl,agsw,k=5,agdl,agsrl) = 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor; folgt: $agSoWiWsA_j(j,agswl,agsw,k=5,agdl,agsrl) = pAgWSchöpf_j(j,agdl,kIKon_{agsw_agd}(j,agswl,agsw,agdl),k=9) - sAgWSchöpf_j(j,agsrl,kIKon_{agsw_agsr}(j,agswl,agsw,agsrl),k=4)$.	163
agÜZeit_j(j,agdl,agd,k=10)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Arbeitsgangdatenvektorenliste (agdl) enthaltenen Arbeitsgangdatenvektors (agd) zu dessen Komponente k=10; dieser Wert repräsentiert die (durchschnittliche) Übergangszeit des zugrunde liegenden Arbeitsgangs zum nächsten Arbeitsgang. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden direkt über Annahmen für die relevanten Prozesse in Form einer nicht negativen reellen Zahl zugeordnet.	78

Offener Funktionsausdruck	Bedeutung	SNr.
agWiIstATA_j(j,agsil,agsi,k=1,agsal,agial)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste arbeitsgangspezifischer Wird-Ist-Abweichungsgrößenvektoren (agwil) enthaltenen arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektors (agwi) zu dessen Komponente k=1 (d.h. der ersten Komponente) unter Berücksichtigung der Liste simulierter arbeitsgangspezifischer Aktionskomponenten (agsal) des betreffenden Prozesses sowie seiner Liste realisierter arbeitsgangspezifischer Aktionskomponenten (agial); dieser Wert repräsentiert die Wird-Ist-Anfangsterminabweichung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 121, 137, 257 und 258 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $sAgAT_j(j,agsal,kIKon_{agwi_agsa3}(j,agwil,agwi,agsal)) = -1$ (d.h., es liegt noch kein simulierter Anfangstermin für den hier relevanten prozessspezifischen Arbeitsgang vor) oder wenn gilt: $iAgAT_j(j,agial,kIKon_{agwi_agia3}(j,agwil,agwi,agial)) = -1$ (d.h., es liegt noch kein realisierter Anfangstermin für den hier relevanten Arbeitsgang vor), dann folgt daraus: $agWiIstATA_j(j,agwil,agwi,k=1,agsal,agial) = 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $agWiIstATA_j(j,agwil,agwi,k=1,agsal,agial) = sAgAT_j(j,agsal,kIKon_{agwi_agsa3}(j,agwil,agwi,agsal)) - iAgAT_j(j,agial,kIKon_{agwi_agia3}(j,agwil,agwi,agial))$.	259

Offener Funktionsausdruck	Bedeutung	SNr.
agWiIstETA_j(j,agwil,agwi, k=2,agsal,agial)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste arbeitsgangspezifischer Wird-Ist-Abweichungsgrößenvektoren (agwil) enthaltenen arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektors (agwi) zu dessen Komponente k=2 (d.h. der zweiten Komponente) unter Berücksichtigung der Liste simulierter arbeitsgangspezifischer Aktionskomponenten (agsal) des betreffenden Prozesses sowie seiner Liste realisierter arbeitsgangspezifischer Aktionskomponenten (agial); dieser Wert repräsentiert die Wird-Ist-Endterminabweichung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 138, 143, 260 und 261 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $sAgET_j(j,agsal,kIKon_{agwi_agsa4}(j,agwil,agwi,agsal)) = -1$ (d.h., es liegt noch kein simulierter Endtermin für den hier relevanten prozessspezifischen Arbeitsgang vor) oder wenn gilt: $iAgET_j(j,agial,kIKon_{agwi_agia4}(j,agwil,agwi,agial)) = -1$ (d.h., es liegt noch kein realisierter Endtermin für den hier relevanten Arbeitsgang vor), dann folgt daraus: $agWiIstETA_j(j,agwil,agwi,k=2,agsal,agial) = 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $agWiIstETA_j(j,agwil,agwi,k=2,agsal,agial) = sAgET_j(j,agsal,kIKon_{agwi_agsa4}(j,agwil,agwi,agsal)) - iAgET_j(j,agial,kIKon_{agwi_agia4}(j,agwil,agwi,agial))$.	262

Offener Funktionsausdruck	Bedeutung	SNr.
agWiIstKA_j(j,agwil,agwi,k=4,agsrl,agirl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste arbeitsgangspezifischer Wird-Ist-Abweichungsgrößenvektoren (agwil) enthaltenen arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektors (agwi) zu dessen Komponente k=4 unter Berücksichtigung der Liste simulierter arbeitsgangspezifischer Reaktionsgrößenvektoren (agsrl) des betreffenden Prozesses sowie seiner Liste realisierter arbeitsgangspezifischer Reaktionsgrößenvektoren (agirl); dieser Wert repräsentiert die Soll-Ist-Kostenabweichung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 147, 212, 263 und 264 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn entweder gilt: $sAgKosten_j(j,agsrl,kIKon_{agwi_agsr}(j,agwil,agwi,agsrl),k=3) = -1$ (d.h., es liegen noch keine Simulationsdaten über die Kosten des hier relevanten prozessspezifischen Arbeitsgangs vor), oder: $iAgKosten_j(j,agirl,kIKon_{agwi_agir}(j,agwil,agwi,agirl),k=3) = -1$ (d.h., es liegen noch keine Istdaten über die Kosten des hier relevanten prozessspezifischen Arbeitsgangs vor), dann folgt daraus: $agWiIstKA_j(j,agwil,agwi,k=4,agsrl,agirl) = 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $agWiIstKA_j(j,agwil,agwi,k=4,agsrl,agirl) := sAgKosten_j(j,agsrl,kIKon_{agwi_agsr}(j,agwil,agwi,agsrl),k=3) - iAgKosten_j(j,agirl,kIKon_{agwi_agir}(j,agwil,agwi,agirl),k=3)$.	266

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
agWiIstOpA_j(j,agwil,agwi,k=3,agsrl,agirl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste arbeitsgangspezifischer Wird-Ist-Abweichungsgrößenvektoren (agwil) enthaltenen arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektors (agwi) zu dessen Komponente k=3 unter Berücksichtigung der Liste simulierter arbeitsgangspezifischer Reaktionsgrößenvektoren (agsrl) des betreffenden Prozesses sowie seiner Liste realisierter arbeitsgangspezifischer Reaktionsgrößenvektoren (agirl); dieser Wert repräsentiert die Wird-Ist-Outputabweichung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 145, 210, 263 und 264 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn entweder gilt: $sAgOutput_j(j,agsrl,kIKon_{agwi_agsr}(j,agwil,agwi,agsrl),k=1) = -1$ (d.h., es liegen noch keine Simulationsdaten über den Output des hier relevanten prozessspezifischen Arbeitsgangs vor), oder: $iAgOutput_j(j,agirl,kIKon_{agwi_agir}(j,agwil,agwi,agirl),k=1) = -1$ (d.h., es liegen noch keine Ist-daten über den Output des hier relevanten prozessspezifischen Arbeitsgangs vor), dann folgt daraus: $agWiIstOpA_j(j,agwil,agwi,k=3,agsrl,agirl) = 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $agSoIstOpA_j(j,agwil,agwi,k=3,agsrl,agirl) = sAgOutput_j(j,agsrl,kIKon_{agwi_agsr}(j,agwil,agwi,agsrl),k=1) - iAgOutput_j(j,agirl,kIKon_{agwi_agir}(j,agwil,agwi,agirl),k=1)$.	265
AgWiIstV(agwi)	Etwas (agwi) ist ein arbeitsgangspezifischer Wird-Ist-Abweichungsgrößenvektor.	253
AgWiIstV_Komp(agwi,k)	Ein arbeitsgangspezifischer Wird-Ist-Abweichungsgrößenvektor (agwi) weist die Komponente (k) auf.	255
AgWiIstVL_AgWiIstV(agwil,agwi)	Eine Liste (agwil) arbeitsgangspezifischer Wird-Ist-Abweichungsgrößenvektoren beinhaltet den arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektor (agwi).	254
AgWiIstVListe(agwil)	Etwas (agwil) ist eine Liste arbeitsgangspezifischer Wird-Ist-Abweichungsgrößenvektoren.	252

Offener Funktionsausdruck	Bedeutung	SNr.
agWiIstWsA_j(j,agwil,agwi,k=5,agsrl,agirl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste arbeitsgangspezifischer Wird-Ist-Abweichungsgrößenvektoren (agwil) enthaltenen arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektors (agwi) zu dessen Komponente k=5 unter Berücksichtigung der Liste simulierter arbeitsgangspezifischer Reaktionsgrößenvektoren (agsrl) des betreffenden Prozesses sowie seiner Liste realisierter arbeitsgangspezifischer Reaktionsgrößenvektoren (agirl); dieser Wert repräsentiert die Wird-Ist-Wertschöpfungsabweichung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 148, 213, 263 und 264 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn entweder gilt: $sAgWSchöpf_j(j,agsrl,kIKon_{agwi_agsr}(j,agwil,agwi,agsrl),k=4) = -1$ (d.h., es liegen noch keine Simulationsdaten über die Wertschöpfung des hier relevanten prozessspezifischen Arbeitsgangs vor) $iAgWSchöpf_j(j,agirl,kIKon_{agwi_agir}(j,agwil,agwi,agirl),k=4) = 0$ (d.h., es liegen noch keine Istdaten über die Wertschöpfung des hier relevanten prozessspezifischen Arbeitsgangs vor), dann folgt daraus: $agWiIstWsA_j(j,agwil,agwi,k=5,agsrl,agirl) = 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $agWiIstWsA_j(j,agwil,agwi,k=5,agsrl,agirl) = sAgWSchöpf_j(j,agsrl,kIKon_{agwi_agsr}(j,agwil,agwi,agsrl),k=4) - iAgWSchöpf_j(j,agirl,kIKon_{agwi_agir}(j,agwil,agwi,agirl),k=4)$.	267
aKEKennz(j)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt das aktuelle Knoteneingangskennzeichen des Prozesses (j). <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden den einzelnen Prozessen (j) jeweils zum Abschluss der originären bzw. aktualisierten Modellierung der Knoteneingangsseite ihres Prozessbeginnknotens in Form der diese Modellierung repräsentierenden Zeichenkette „EKon“ oder „EEx“ zugewiesen.	40

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
AKEKennzeichen(ake)	Etwas (ake) ist ein aktuelles Knoteneingangskennzeichen.	39
AktAnpassVar(aav)	Etwas (aav) ist eine aktuelle Anpassungsvariante.	26
aktAnzNegVeri_a(j,pr,avl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) die aktuelle Anzahl der zur (Prozess-) Regel (pr) mit der Regelanwendungsvoraussetzungsliste (avl) vorliegenden unzureichenden Verifikationsergebnisse. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe des in SNr. 325 spezifizierten Funktionsausdrucks und den in SNr. 328 und 329 spezifizierten wff festgelegt. Dabei werden zwei Fälle unterschieden. unter der Bedingung, dass die Antezedenz der Implikation der unter SNr. 326 spezifizierten wff erfüllt ist, wie folgt aktualisiert: Fall 1): liegt vor, wenn die Antezedenz der Implikation der unter SNr. 328 spezifizierten wff erfüllt ist, in diesem Fall gilt: $\text{aktAnzNegVeri_a(j,pr,avl)} := \text{initAnzNegVeri_a(j,pr,avl)} + 1.$ Fall 2): liegt vor, wenn die Antezedenz der Implikation der unter SNr. 329 spezifizierten wff erfüllt ist, in diesem Fall gilt: $\text{aktAnzNegVeri_a(j,pr,avl)} := 0.$	326
AktCZeitstufe(acs)	Etwas (acs) ist eine aktuelle Controlling-Zeitstufe.	96

Offener Funktionsausdruck	Bedeutung	SNr.
alokG(j,vpnl,vpn)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt das aktuelle lokale Gewicht des Prozesses (j), ausgehend von seinem unmittelbaren Vorgängerprozess mit der Prozessnummer (vpn), die in der Vorgängerprozessnummernliste (vpn�) enthalten ist. <u>Funktionsgleichung:</u> Bei der Funktionswertermittlung werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn (vpn) die Individuenkonstante None aufweist (d.h. (j) weist keinen unmittelbaren Vorgänger auf) oder wenn der zu (vpn) gehörende Vorgängerprozess mit der aktuellen Phase „Ausführung“, d.h. aph=A, gekennzeichnet ist; dann gilt: alokG(j):=1. Fall 2): liegt vor, wenn der zu (vpn) gehörende Vorgängerprozess mit der aktuellen Phase „Entscheidung“, d.h. aph=E, gekennzeichnet ist; dann gilt: alokG(j):= x, mit $0 < x < 1$; x ist dabei von der oder den für (j) verantwortlichen Prozess- oder Projektinstitution bzw. -institutionen entsprechend festzulegen.	38
anzEProzMinLokG(cs)	<u>Funktionsbeschreibung:</u> Diese Funktion gibt zu dem Controlling-Zeitpunkt, welcher der Controlling-Zeitstufe (cs) zugeordnet ist, die Obergrenze der in einem geplanten Prozesspfad eines OEN enthaltenen Entscheidungsprozesse¹⁰ wieder, die dadurch gekennzeichnet sind, dass sie jeweils mit dem unmittelbaren Nachfolgeprozess (exklusiv disjunkt) verknüpft sind, dessen lokales Eintrittsgewicht im Vergleich zu den alternativen Nachfolgeprozessen minimal ist. <u>Funktionsgleichung:</u> Der jeweilige Funktionswert entspricht dem im zugrunde liegenden Controlling-Zeitpunkt (von dafür verantwortlichen Steuerungsinstitutionen) zu ermittelnden Wert $nemlg_{j_p} \in \mathbb{N}_{>0}$, (mit $j_p = 1, \dots, J_p \in \mathbb{N}_{>0}$) für dessen Indexwert gilt: $j_p = cs$, d.h.: anzEProzMinLokG(cs) := $nemlg_{cs}$.	382

¹⁰ Wie bei SNr. 379 werden als Entscheidungsprozesse vereinfachend die (Unter-) Prozesse bezeichnet, deren aktuelle Phase (gemäß $dp_{24}(up)$) „Entscheidung“ ist.

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
anzKomp_{inpb}(inpb)	<u>Funktionsbeschreibung</u>: Diese Funktion gibt die Anzahl der Komponenten des realisierten entwicklungsprojektspezifischen (Ist-) Nutzungspotenzialbestandsvektors (inpb) wieder. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden realisierten entwicklungsprojektspezifischen (Ist-) Nutzungspotenzialbestandsvektoren (inpb) über Annahmen im jeweiligen Entwicklungsprojekt jeweils in Form einer natürlichen Zahl zugeordnet.	340
anzKomp_{pnpb}(pnpb)	<u>Funktionsbeschreibung</u>: Diese Funktion gibt die Anzahl der Komponenten des geplanten entwicklungsprojektspezifischen Nutzungspotenzialbestandsvektors (pnpb) wieder. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden geplanten entwicklungsprojektspezifischen Nutzungspotenzialbestandsvektoren (pnpb) über Annahmen im jeweiligen Entwicklungsprojekt jeweils in Form einer natürlichen Zahl zugeordnet.	336
anzKomp_{snpb}(snpb)	<u>Funktionsbeschreibung</u>: Diese Funktion gibt die Anzahl der Komponenten des simulierten entwicklungsprojektspezifischen (Wird-) Nutzungspotenzialbestandsvektors (snpb) wieder. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden simulierten entwicklungsprojektspezifischen (Wird-) Nutzungspotenzialbestandsvektoren (snpb) über Annahmen im jeweiligen Entwicklungsprojekt jeweils in Form einer natürlichen Zahl zugeordnet.	338
APhase(aph)	Etwas (aph) ist eine aktuelle Prozessphase.	4
aPhase(j)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt die aktuelle Phase des Prozesses (j). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den einzelnen Prozessen (j) zum Abschluss der originären bzw. aktualisierten Phasenspezifikation in Form des diese Phase kennzeichnenden Buchstabens („E“ oder „A“) zugewiesen.	6

Offener Funktionsausdruck	Bedeutung	SNr.
aPhase_{vpnl_vpn}(vpnl,vpn)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für die zu einer Vorgängerprozessnummernliste (vpnl) gehörende Vorgängerprozessnummer (vpn) die aktuelle Phase des zugehörigen Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mit Hilfe der aPhase(j) ermittelt (siehe SNr. 6). Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, falls die Vorgängerprozessnummernliste (vpnl) den Eintrag vpn=None aufweist; sodann gilt: aPhase_{vpnl_vpn}(vpnl,None):= None. Fall 2) liegt vor, falls Fall 1 nicht zutrifft, sodann gilt mit j=vpn: aPhase_{vpnl_vpn}(vpnl,vpn):= aPhase(vpn)¹¹.	34
aPLaufNr(acs)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt die aktuelle Planungslaufnummer zur aktuellen Controlling-Zeitstufe (acs). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden aktuellen Controlling-Zeitstufen (acs) anhand dazu vorliegender Planungsinformationen im relevanten (Teil-) Projektkontext in Form natürlicher Zahlen einschließlich der Zahl 0 direkt zugeordnet. Sofern zu (acs) noch kein (aktueller) Planungslauf vorliegt, gilt: aPLaufNr(acs):=0; andernfalls wird aPLaufNr(acs) die dann aktuelle Planungslaufnummer zugewiesen.	97

¹¹ Diese Funktion ist insbesondere für die Ermittlung der Funktionswerte der unter SNr. 38 spezifizierten Funktion aLokG(j) relevant.

Offener Funktionsausdruck	Bedeutung	SNr.
aProzAbfolgeNr(j, aPLaufNr(acs))	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt die aktuelle Prozessabfolgenummer des Prozesses (j), die im aktuellen Planungslauf mit der Laufnummer gemäß dem Wert der in SNr. 95 spezifizierten Funktion aPLaufNr(acs) vorliegt. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden anhand der für die jeweils aktuelle Controlling-Zeitstufe (acs) im dazu vorliegenden Planungslauf vorhandenen Planungsinformationen zugeordnet. Dabei werden die nachfolgenden drei Fälle unterschieden¹². Fall 1): liegt vor, wenn aPLaufNr(acs) = 0 ist, d.h. noch kein Planungslauf vorliegt, oder wenn der zugrunde liegende Prozess (j) noch nicht eingeplant wurde; sodann gilt: $\text{aProzAbfolgeNr}(j, \text{aPLaufNr}(\text{acs})) = -1$. Fall 2): liegt vor, wenn der betreffende Prozess (j) im vorliegenden Planungslauf nicht mehr berücksichtigt ist; dann gilt: $\text{aProzAbfolgeNr}(j, \text{aPLaufNr}(\text{acs})) = -2$. Fall 3): liegt vor, wenn der zugrunde liegende Prozess (j) im vorliegenden Planungslauf mit einer entsprechenden Prozessabfolgenummer $k \in \mathbb{N}_{>0}$ eingeplant ist; dann gilt entsprechend: $\text{aProzAbfolgeNr}(j, \text{aPLaufNr}(\text{acs})) = k, k \in \mathbb{N}_{>0}$.	98
atermin_{j_lat_at}(j,lat,at)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Termin der zu seiner Liste relevanter Aktionszeitpunktpositionen (lat) gehörenden Aktionszeitpunktposition (at). <u>Funktionsgleichung</u>: Die Funktionswerte werden den einzelnen relevanten Aktionszeitpunktpositionen (at) der betreffenden Liste (lat) über Annahmen für den zugrunde liegenden Prozess (j) im jeweils relevanten (Teil-) Projektkontext in Form einer nicht negativen reellen Zahl zugeordnet; für den Fall, dass (lat) die unechte Aktionszeitpunktposition at=None beinhaltet, gilt: $\text{atermin}_{j_lat_at}(j, \text{lat}, \text{at}) = 0$.	56

¹² Vgl. hierzu auch die Ausführungen zur Spezifikation des Wertebereichs des Parameters $dp_{38}(up)$. Die erforderlichen Planungsläufe können insbesondere durch projektinterne Planungsprozesse hervorgerufen werden, d.h. für Prozesse, die durch das geschlossene Atom $\text{Proz_Prozessart}(j, \text{BS_Determ})$ gemäß SNr. 3 und der Teilmenge PZK_1 ($PZK_1 \subset PZK$, gemäß $dp_2(up)$) gekennzeichnet sind.

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
AVL_AV_KA(avl,av,ka)	Die in einer Regelanwendungsvoraussetzungsliste (avl) enthaltene Anwendungsvoraussetzung weist die Kritizitätsausprägung (ka) auf.	315
AVPRegel(av)	Etwas (av) ist eine (Prozess-) Regelanwendungsvoraussetzung.	16
AVPRListe(avl)	Etwas (avl) ist eine (Prozess-) Regelanwendungsvoraussetzungsliste	17
AVPRListe_AVPR(avl,av)	Eine (Prozess-) Regelanwendungsvoraussetzungsliste (avl) beinhaltet die Prozessregelanwendungsvoraussetzung (av).	18
aZusatzK(j)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt das aktuelle Zusatzkennzeichen des Prozesses (j). <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden den einzelnen Prozessen (j) jeweils zum Abschluss der originären bzw. aktualisierten Spezifikation der Zusatzkennzeichnung in Form der Zeichenketten („Gin“, „Oin“, „Ex“, „Unb“) zugewiesen.	8
Beitragspos(bp)	Etwas (bp) ist eine Planbeitragsposition.	21
bErfüllung_{j_pr_avl_av}(j,pr,avl,av)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert, der die Erfüllung oder Nichterfüllung der in der (Prozess-) Regelanwendungsvoraussetzungsliste (avl) enthaltenen (Prozess-) Regelanwendungsvoraussetzung (av) repräsentiert. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden gemäß dem Ergebnis der jeweils durchzuführenden Prüfung der Anwendungsvoraussetzung zugewiesen. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn die betreffende Anwendungsvoraussetzung erfüllt ist, in diesem Fall gilt: $bErfüllung_{j_pr_avl_av}(j,pr,avl,av) = 1$. Fall 2): liegt vor, wenn Fall 1) nicht vorliegt, d.h. die betreffende Anwendungsvoraussetzung nicht erfüllt ist; sodann gilt: $bErfüllung_{j_pr_avl_av}(j,pr,avl,av) = 0$.	316
BPPZL_BP_PZ(bppzl,bp,pz)	Eine (bppzl) weist zu einer Beitragsposition (bp) das Prozessziel (pz) auf.	23
BPPZListe(bppzl)	Etwas (bppzl) ist eine Liste von Beitragspositionen, die Projektzielen zugeordnet sind.	22

Offener Funktionsausdruck	Bedeutung	SNr.
czeit_{j_aav}(j,aav)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Controlling-Zeitpunkt der Spezifikation seiner aktuellen Anpassungsvariante (aav). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den einzelnen Prozessen (j) jeweils zum Abschluss ihrer durch (aav) geprägten originären bzw. aktualisierten Spezifikation in Form des dann vorliegenden Controlling-Zeitpunktes zugeordnet.	28
czeit_{j_ake}(j,ake)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt zum Prozess (j) den Controlling-Zeitpunkt der Spezifikation seines aktuellen Knoteneingangskennzeichens (ake). <u>Funktionsgleichung</u>: Die Funktionswerte werden den einzelnen Prozessen (j) jeweils zum Abschluss ihrer durch (ake) repräsentierten originären bzw. aktualisierten Modellierung des Knoteneingangsseite des zugehörigen Prozessbeginnknottes in Form des dann vorliegenden Controlling-Zeitpunktes zugeordnet.	41
czeit_{j_aph}(j,aph)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Controlling-Zeitpunkt, welcher der Spezifikation seiner aktuellen Phase (aph) zugrunde liegt. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den einzelnen Prozessen (j) jeweils zum Abschluss ihrer durch (aph) gekennzeichneten originären bzw. aktualisierten Spezifikation in Form des dann vorliegenden Controlling-Zeitpunktes zugeordnet.	7
czeit_{j_azk}(j,azk)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt zum Prozess (j) den Controlling-Zeitpunkt der Spezifikation seines aktuellen Zusatzkennzeichens (azk). <u>Funktionsgleichung</u>: Die Funktionswerte werden den einzelnen Prozessen (j) jeweils zum Abschluss ihrer durch (azk) gekennzeichneten originären bzw. aktualisierten Spezifikation in Form des dann vorliegenden Controlling-Zeitpunktes zugeordnet.	9

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
cZeitpunkt(cs)	<u>Funktionsbeschreibung</u>: Diese Funktion gibt den Controlling-Zeitpunkt der Controlling-Zeitstufe (cs) wieder. <u>Funktionsgleichung</u>: Ausgehend von den im zugrunde liegenden Entwicklungsprojekt von den projektverantwortlichen Institutionen festgelegten Controlling-Zeitpunkten $tp_{j_{tp}} \in \mathbb{R}_+$ (mit $j_{tp} = 1, \dots, J_{tp} \in \mathbb{N}_{>0}$) wird jeweils derjenige Controlling-Zeitpunkt der Controlling-Zeitstufe (cs) zugeordnet, für dessen Indexwert gilt: $j_{tp} = cs$, d.h.: $cZeitpunkt(cs) := tp_{cs}$.	379
CZeitstufe(cs)	Etwas (cs) ist eine Controlling-Zeitstufe.	377
dMinLokG(cs)	<u>Funktionsbeschreibung</u>: Diese Funktion gibt zu dem Controlling-Zeitpunkt, welcher der Controlling-Zeitstufe (cs) zugeordnet ist, den Durchschnitt der Minimalwerte der lokalen Eintrittsgewichte der mit Entscheidungsprozessen¹³ als unmittelbaren Vorgängerprozessen jeweils verknüpften (Unter-) Prozesse wieder. <u>Funktionsgleichung</u>: Der jeweilige Funktionswert entspricht dem im zugrunde liegenden Controlling-Zeitpunkt (von dafür verantwortlichen Steuerungsinstitutionen) zu ermittelnden Wert $dmlg_{j_{tp}} \in]0, 1[$, (mit $j_{tp} = 1, \dots, J_{tp} \in \mathbb{N}_{>0}$), für dessen Indexwert gilt: $j_{tp} = cs$, d.h.: $dMinLokG(cs) := dmlg_{cs}$.	381
efDauer_j(j)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt die erwartete bzw. fixierte Dauer des Prozesses (j). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden Prozessen (j) über Annahmen im relevanten (Teil-) Projektkontext jeweils in Form einer nicht negativen reellen Zahl direkt zugeordnet.	93

¹³ Als Entscheidungsprozesse werden vereinfachend die (Unter-) Prozesse bezeichnet, deren aktuelle Phase (gemäß $dp_{24}(up)$) „Entscheidung“ ist.

Offener Funktionsausdruck	Bedeutung	SNr.
ef₁(j,npnl,npn,uzl,uz)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt den Ergebnisfunktionswert vom Typ 1 für den zur Nachfolgerprozessnummernliste (npnl) von Prozess (j) gehörenden Nachfolgeprozess (npn) unter Berücksichtigung des zu seiner (Entscheidungs-) Umfeldzustandsliste (uzl) gehörenden Umweltzustands (uz). <u>Funktionsgleichung:</u> Die Funktionswerte werden über Annahmen für die relevanten Nachfolgeprozesse (npn) im jeweils relevanten (Teil-) Projektkontext in Form nicht negativer reeller Zahlen zugeordnet; falls (npn) oder (uz) die Individuenkonstante „None“ aufweisen, gilt: $ef_1(j,npnl,npn,uzl,uz):=0$.	57
ef₂(j,npnl,npn,uzl,uz, wert_{j_przl_prz}(j,przl,pzl))	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt den Ergebnisfunktionswert vom Typ 2 für den zur Nachfolgerprozessnummernliste (npnl) von Prozess (j) gehörenden Nachfolgeprozess (npn) unter Berücksichtigung des zur (Entscheidungs-) Umfeldzustandsliste (uzl) von Prozess (j) gehörenden Umweltzustands (uz) und des Funktionswerts des zur Prozesszielliste (przl) von Prozess (j) gehörenden Prozessziels (prz) gemäß SNr. 55. <u>Funktionsgleichung:</u> Die Funktionswerte werden über Annahmen für die relevanten Nachfolgeprozesse (npn) im jeweils relevanten (Teil-) Projektkontext in Form nicht negativer reeller Zahlen zugeordnet; falls (npn) oder auch (uz) die Individuenkonstante „None“ aufweisen, gilt: $ef_2(j,npnl,npn,uzl,uz, wert_{j_przl_prz}(j,przl,pzl)):=0$.	58

Offener Funktionsausdruck	Bedeutung	SNr.
ef₃(j,npnl,npn,uzl,uz, wert_{j_przl_prz}(j,przl,pzl), atermin_{j_lat_at}(j,lat,at))	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt den Ergebnisfunktionswert vom Typ 3 für den zur Nachfolgerprozessnummernliste (npnl) von Prozess (j) gehörenden Nachfolgeprozess (npn) unter Berücksichtigung des zur (Entscheidungs-) Umfeldzustandsliste (uzl) von Prozess (j) gehörenden Umweltzustands (uz), des Funktionswerts des zur Prozesszielliste (przl) von Prozess (j) gehörenden Prozessziels (prz) gemäß SNr. 55 und des Funktionswerts der zur Liste relevanter Aktionszeitpunktpositionen (lat) von Prozess (j) gehörenden Aktionszeitpunktposition (at) gem. SNr. 56. <u>Funktionsgleichung:</u> Die Funktionswerte werden über Annahmen für die relevanten Nachfolgeprozesse (j) im jeweils relevanten (Teil-) Projektkontext in Form nicht negativer reeller Zahlen zugeordnet; falls (npn), (uz) oder auch (at) die Individuenkonstante „None“ aufweisen, gilt: $\text{ef}_3(j, npnl, npn, uzl, uz, \text{wert}_{j_przl_prz}(j, przl, prz), \text{atermin}_{j_lat_at}(j, lat, at)) = 0$.	59
fATermin_j(j)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt den frühesten Anfangstermin des Prozesses (j). <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden den betreffenden Prozessen über Annahmen im relevanten (Teil-) Projektkontext jeweils in Form einer nicht negativen reellen Zahl direkt zugeordnet.	91
fETermin_j(j)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt den frühesten Endtermin des Prozesses (j). <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden den entsprechenden Prozessen (j) über Annahmen im relevanten (Teil-) Projektkontext jeweils in Form einer nicht negativen reellen Zahl direkt zugeordnet, wobei folgende Bedingung gilt: $\text{fETermin}(j) \geq \text{spATermin}(j)$.	94

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
iAgAT_j(j,agial,agia₃)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den realisierten Arbeitsganganfangstermin zu der in seiner Liste der realisierten arbeitgangspezifischen Aktionskomponenten (agial) enthaltenen realisierten Arbeitsganganfangsterminposition (agia₃). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der gegebenenfalls für die jeweils relevanten prozessspezifischen Arbeitsgänge vorliegenden Prozessrealisationsinformationen zugeordnet¹⁴. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn (noch) kein realisierter Anfangstermin für den hier relevanten prozessspezifischen Arbeitsgang vorhanden ist; in diesem Fall gilt: $\text{iAgAT}_j(j, \text{agial}, \text{agia}_3) := -1.$ Fall 2): liegt vor, wenn ein realisierter Anfangstermin für den hier relevanten prozessspezifischen Arbeitsgang vorhanden ist; sodann gilt: $\text{iAgAT}_j(j, \text{agial}, \text{agia}_3) := t, \quad t \in \mathbb{R}_+.$	137
IAgATPos(agia₃)	Etwas (agia ₃) ist eine realisierte arbeitgangspezifische Anfangsterminposition.	131

¹⁴ Die hier betrachteten Funktionswerte repräsentieren jeweils den realisierten Arbeitsganganfangstermin des Arbeitsgangs j , $\text{agia}_{j,3}$, innerhalb der Spezifikation des Parameters $\text{ap}_5(\text{up})$.

Offener Funktionsausdruck	Bedeutung	SNr.
iAgAusschuss_j(j,agirl,agir,k=2)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste realisierter arbeitgangspezifischer Reaktionsgrößenvektoren (agirl) enthaltenen realisierten arbeitgangspezifischen Reaktionsgrößenvektors (agir) zu dessen Komponente k=2; dieser Wert repräsentiert den gegebenenfalls vorliegenden realisierten Ausschuss des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten prozessspezifischen Arbeitsgänge gegebenenfalls vorliegenden Realisationsinformationen festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch kein echter realisierter arbeitgangspezifischer Reaktionsgrößenvektor in der betreffenden Liste (agirl) vorhanden ist [(agir) ist dabei mit der Individuenkonstante „None“ belegt] oder noch keine Istdaten über den realisierten Ausschuss (kurz <i>ra</i>) für den hier relevanten Reaktionsgrößenvektor gegeben sind; sodann gilt: $\text{iAgAusschuss}_j(j,\text{agirl},\text{agir},k=2) := -1.$ Fall 2): liegt vor, wenn Istdaten für den hier relevanten arbeitgangspezifischen Reaktionsgrößenvektor über den entsprechend realisierten Ausschuss <i>ra</i> gegeben sind, sodann gilt: $\text{iAgAusschuss}_j(j,\text{agirl},\text{agir},k=2) := ra,$ mit $\text{agir} \in \text{AGIR} \setminus \{\text{None}\}$, $ra \in \mathbb{N}$.	211

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
iAgET_j(j, agial, agia₄)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den realisierten Arbeitsgangendtermin zu der in seiner Liste der realisierten arbeitgangspezifischen Aktionskomponenten (agial) enthaltenen realisierten Arbeitsgangendterminposition (agia₄). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der gegebenenfalls für die jeweils relevanten prozessspezifischen Arbeitsgänge vorliegenden Prozessrealisationsinformationen zugeordnet¹⁵. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn (noch) kein realisierter Endtermin für den hier relevanten prozessspezifischen Arbeitsgang vorhanden ist; in diesem Fall gilt: $\text{iAgET}_j(j, \text{agial}, \text{agia}_4) = -1.$ Fall 2): liegt vor, wenn ein realisierter Endtermin für den hier relevanten prozessspezifischen Arbeitsgang vorhanden ist; sodann gilt: $\text{iAgET}_j(j, \text{agial}, \text{agia}_4) = t, \quad t \in \mathbb{R}_+.$	138
IAgETPos(agia₄)	Etwas (agia ₄) ist eine realisierte arbeitgangspezifische Endterminposition.	133

¹⁵ Die hier betrachteten Funktionswerte repräsentieren jeweils den realisierten Arbeitsgangendtermin des Arbeitsgangs j , $\text{agia}_{j,4}$, innerhalb der Spezifikation des Parameters $\text{ap}_5(\text{up})$.

Offener Funktionsausdruck	Bedeutung	SNr.
iAgKosten_j(j,agirl,agir,k=3)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste realisierter arbeitgangspezifischer Reaktionsgrößenvektoren (agirl) enthaltenen realisierten arbeitgangspezifischen Reaktionsgrößenvektors (agir) zu dessen Komponente k=3; dieser Wert repräsentiert die gegebenenfalls vorliegenden realisierten Kosten des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten prozessspezifischen Arbeitsgänge gegebenenfalls vorliegenden Realisationsinformationen festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch kein echter realisierter arbeitgangspezifischer Reaktionsgrößenvektor in der betreffenden Liste (agirl) vorhanden ist [(agir) ist dabei mit der Individuenkonstante „None“ belegt] oder noch keine Istdaten über die realisierten Kosten (kurz rk) für den hier relevanten Reaktionsgrößenvektor gegeben sind; in diesem Fall gilt: $\text{iAgKosten}_j(j,\text{agirl},\text{agir},k=3) := -1.$ Fall 2): liegt vor, wenn Istdaten für den hier relevanten arbeitgangspezifischen Reaktionsgrößenvektor über die entsprechend realisierten Kosten rk gegeben sind, sodann gilt: $\text{iAgKosten}_j(j,\text{agirl},\text{agir},k=3) := rk,$ mit $\text{agir} \in \text{AGIR} \setminus \{\text{None}\}$, $rk \in \mathbb{R}_+$.	212

Offener Funktionsausdruck	Bedeutung	SNr.
iAgOutput_j(j,agirl,agir,k=1)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste realisierter arbeitgangspezifischer Reaktionsgrößenvektoren (agirl) enthaltenen, realisierten arbeitgangspezifischen Reaktionsgrößenvektors (agir) zu dessen Komponente k=1 (d.h. der ersten Komponente); dieser Wert repräsentiert den gegebenenfalls vorliegenden realisierten (Ist-) Output des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten prozessspezifischen Arbeitsgänge gegebenenfalls vorliegenden Realisationsinformationen festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch kein echter realisierter arbeitgangspezifischer Reaktionsgrößenvektor in der betreffenden Liste (agirl) vorhanden ist [(agir) ist dabei mit der Individuenkonstante „None“ belegt] oder noch keine Istdaten über den realisierten Output (kurz <i>ro</i>) für den hier relevanten Reaktionsgrößenvektor gegeben sind; sodann gilt: $\text{iAgOutput}_j(j,\text{agirl},\text{agir},k=1) := -1.$ Fall 2): liegt vor, wenn Istdaten für den hier relevanten arbeitgangspezifischen Reaktionsgrößenvektor über den entsprechend realisierten Output <i>ro</i> gegeben sind; sodann gilt: $\text{iAgOutput}_j(j,\text{agirl},\text{agir},k=1) := ro,$ mit $\text{agir} \in \text{AISR} \setminus \{\text{None}\}, ro \in \mathbb{N}.$	210

Offener Funktionsausdruck	Bedeutung	SNr.
iAgWSchöpf_j(j,agirl,agir,k=4)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste realisierter arbeitgangspezifischer Reaktionsgrößenvektoren (agirl) enthaltenen realisierten arbeitgangspezifischen Reaktionsgrößenvektors (agir) zu dessen Komponente k=4; dieser Wert repräsentiert die gegebenenfalls vorliegende realisierte Wertschöpfung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten prozessspezifischen Arbeitsgänge gegebenenfalls vorliegenden Realisationsinformationen festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch kein echter realisierter arbeitgangspezifischer Reaktionsgrößenvektor in der betreffenden Liste (agirl) vorhanden ist [(agir) ist dabei mit der Individuenkonstante „None“ belegt] oder noch keine Istdaten über die realisierte Wertschöpfung (kurz rw) für den hier relevanten Reaktionsgrößenvektor gegeben sind; sodann gilt: $\text{iAgWSchöpf}_j(j, \text{agirl}, \text{agir}, k=4) := -1.$ Fall 2): liegt vor, wenn Istdaten für den hier relevanten arbeitgangspezifischen Reaktionsgrößenvektor über die entsprechend realisierte Wertschöpfung rw gegeben sind, sodann gilt: $\text{iAgWSchöpf}_j(j, \text{agirl}, \text{agir}, k=4) := rw,$ mit $\text{agir} \in \text{AGSR} \setminus \{\text{None}\}$, $rw \in \mathbb{R}_+$.	213
iATermin_j(j)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt den realisierten Anfangstermin bzw. Ist-Anfangstermin des Prozesses (j). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden Prozessen (j) anhand der im relevanten (Teil-) Projektkontext verfügbaren Prozessrealisationsinformationen direkt zugeordnet; dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn (noch) kein Ist-Anfangstermin für den Prozess (j) vorhanden ist; sodann gilt: $\text{iATermin}_j(j) := -1$. Fall 2): liegt vor, wenn ein realisierter Anfangstermin für den Prozess (j) vorhanden ist; sodann gilt: $\text{iATermin}_j(j) := t$, $t \in \mathbb{R}_+$.	124

Offener Funktionsausdruck	Bedeutung	SNr.
iAusschuss_j(j,ir,k=2)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines realisierten Reaktionsgrößenvektors (ir) zu dessen Komponente k=2; dieser Wert repräsentiert den gegebenenfalls vorliegenden realisierten Ausschuss des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten Prozesse gegebenenfalls vorliegenden Realisationsinformationen festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Istdaten über den realisierten Ausschuss (kurz <i>ra</i>) für den hier relevanten Reaktionsgrößenvektor gegeben sind; sodann gilt: iAusschuss_j(j,ir,k=2):= -1. Fall 2): liegt vor, wenn Istdaten für den hier relevanten Reaktionsgrößenvektor über den entsprechend realisierten Ausschuss <i>ra</i> gegeben sind, sodann gilt: iAusschuss_j(j,ir,k=2):= <i>ra</i>, mit $ra \in \mathbb{N}_0$.	236

Offener Funktionsausdruck	Bedeutung	SNr.
ibwert_{j_pr_bppzl}(j,pr,bppzl,bp,pz)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt den realisierten Wert der Beitragsposition (bp), die dem Projektziel (pz) zugeordnet ist und in der Liste Projektzielen zugeordneter Beitragspositionen (bppzl) enthalten ist, welche für die Prozessregel (pr) spezifiziert ist, die im Prozess (j) zur Anwendung vorgesehen ist. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden über entsprechende Annahmen für die Anwendung der Prozessregel (pr) im Prozess (j) im jeweiligen Projektkontext zugeordnet. Dabei werden drei Fälle unterschieden. Fall 1): liegt vor, falls die Individuenkonstante „None“ für (pz) gegeben ist, sodann gilt: $\text{ibwert}_{j_pr_bppzl}(j,pr,bppzl,bp,pz) := \text{nd}$ („nd“ steht hier für „nicht definiert“). Fall 2) liegt vor, falls Fall 1) nicht zutrifft und noch keine Realisationsdaten über den Wert der zugrunde liegenden Beitragsposition gegeben sind; sodann gilt: $\text{ibwert}_{j_pr_bppzl}(j,pr,bppzl,bp,pz) := \text{nv}$ („nv“ steht hier für „no value“ bzw. „kein Wert“). Fall 3) liegt vor, falls Fall 1) nicht zutrifft und Realisationsdaten über den Wert der zugrunde liegenden Beitragsposition gegeben sind; sodann wird als Wert, je nach Projektziel, entweder eine reelle Zahl oder ein alphanumerischer Ausdruck zugewiesen.	283
iETermin_j(j)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt den realisierten Endtermin des Prozesses (j). <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden den entsprechenden Prozessen (j) anhand der im relevanten (Teil-) Projektkontext verfügbaren Prozessrealisationsinformationen direkt zugeordnet. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn (noch) kein realisierter Endtermin für den Prozess (j) vorhanden ist; sodann gilt: $\text{iETermin}_j(j) := -1$. Fall 2): liegt vor, wenn ein realisierter Endtermin für den Prozess (j) vorhanden ist; sodann gilt: $\text{iETermin}_j(j) := t, \quad t \in \mathbb{R}_+$.	139

Offener Funktionsausdruck	Bedeutung	SNr.
iKosten_j(j,ir,k=3)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines realisierten Reaktionsgrößenvektors (ir) zu dessen Komponente k=3; dieser Wert repräsentiert die gegebenenfalls vorliegenden realisierten Kosten des zugrunde liegenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten Prozesse gegebenenfalls vorliegenden Realisationsinformationen festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Istdaten über die realisierten Kosten (kurz rk) für den hier relevanten Reaktionsgrößenvektor gegeben sind; sodann gilt: iKosten_j(j,ir,k=3):= -1. Fall 2): liegt vor, wenn Istdaten für den hier relevanten Reaktionsgrößenvektor über die entsprechenden realisierten Kosten rk gegeben sind, sodann gilt: iKosten_j(j,ir,k=3):= rk, mit $rk \in \mathbb{R}_+$.	237
initAnzNegVeri_a(j,pr,avl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) mit der (Prozess-) Regel (pr) und der Regelanwendungsvoraussetzungsliste (avl) die vor einem diesbezüglich durchzuführenden Verifikationsversuch zu berücksichtigende Anzahl entsprechender, bereits erfolglos durchgeführter Verifikationsversuche. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden vor der Durchführung des betreffenden Verifikationsversuchs entsprechend der dokumentierten Anzahl der betreffenden in einem Entwicklungsprojekt bereits durchgeführten erfolglosen Verifikationsergebnisse in Form einer natürlichen Zahl zugewiesen¹⁶.	325

¹⁶ Dabei wird unterstellt, dass die Anwendung der Prädikatenlogik innerhalb einzelner Simulationsläufe oder realiter durchzuführender Analysen vorab jeweils durch die Spezifikation abgeschlossener Atome sowie abgeschlossener Terme initialisiert wird. Hinsichtlich der in einer entsprechenden Anwendung der Prädikatenlogik durchzuführenden Verifikationsversuche wird ferner unterstellt, dass je zugrunde liegendem Basissteuerungsprozess nur ein Verifikationsversuch durchgeführt wird. Die jeweiligen Verifikationsversuchsergebnisse werden dann für weitere, darauf folgende Anwendungen der Prädikatenlogik dokumentiert.

Offener Funktionsausdruck	Bedeutung	SNr.
initBestand_{inpb_k}(inpb,k)	<u>Funktionsbeschreibung</u>: Diese Funktion gibt den Initialwert der Komponente (k) des realisierten entwicklungsprojektspezifischen (Ist-) Nutzungspotenzialbestandsvektors (inpb) wieder. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden Komponenten (k) der realisierten entwicklungsprojektspezifischen (Ist-) Nutzungspotenzialvektoren (inpb) über Annahmen im jeweiligen Entwicklungsprojekt jeweils in Form einer natürlichen Zahl oder der Zahl „0“ zugeordnet.	343
initBestand_{ipb_k}(ipb,k)	<u>Funktionsbeschreibung</u>: Diese Funktion gibt den Initialwert der Komponente (k) des realisierten entwicklungsprojektspezifischen (Ist-) Potenzialbestandsvektors (ipb) wieder. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 340, 343, 355, 373 und 374 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $k \leq \text{anzKomp}_{\text{inpb}}(k \text{Kon}_{\text{ipb_inpb}}(\text{ipb}))$; daraus folgt: $\text{initBestand}_{\text{ipb_k}}(\text{ipb},k) := \text{initBestand}_{\text{inpb_k}}(k \text{Kon}_{\text{ipb_inpb}}(\text{ipb}),k).$ Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $\text{initBestand}_{\text{ipb_k}}(\text{ipb},k) := \text{initBestand}_{\text{ivpb_k}}(k \text{Kon}_{\text{ipb_ivpb}}(\text{ipb}), k - \text{anzKomp}_{\text{inpb}}(k \text{Kon}_{\text{ipb_inpb}}(\text{ipb}))).$	375
initBestand_{ivpb_k}(ivpb,k)	<u>Funktionsbeschreibung</u>: Diese Funktion gibt den Initialwert der Komponente (k) des realisierten entwicklungsprojektspezifischen (Ist-) Verbrauchspotenzialbestandsvektors (ivpb) wieder. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden Komponenten (k) der realisierten entwicklungsprojektspezifischen (Ist-) Verbrauchspotenzialvektoren (ivpb) über Annahmen im jeweiligen Entwicklungsprojekt jeweils in Form einer nicht negativen reellen Zahl zugeordnet.	355

Offener Funktionsausdruck	Bedeutung	SNr.
initBestand_{pnpb,k}(pnpb,k)	<u>Funktionsbeschreibung:</u> Diese Funktion gibt den Initialwert der Komponente (k) des geplanten entwicklungsprojektspezifischen Nutzungspotenzialbestandsvektors (pnpb) wieder. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden den entsprechenden Komponenten (k) der geplanten entwicklungsprojektspezifischen Nutzungspotenzialvektoren (pnpb) über Annahmen im jeweiligen Entwicklungsprojekt jeweils in Form einer natürlichen Zahl oder der Zahl „0“ zugeordnet.	341
initBestand_{ppb,k}(ppb,k)	<u>Funktionsbeschreibung:</u> Diese Funktion gibt den Initialwert der Komponente (k) des geplanten entwicklungsprojektspezifischen Potenzialbestandsvektors (ppb) wieder. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 336, 341, 353, 360 und 361 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $k \leq \text{anzKomp}_{\text{pnpb}}(\text{kIKon}_{\text{ppb_pnpb}}(\text{ppb}))$; daraus folgt: $\text{initBestand}_{\text{ppb_k}}(\text{ppb},k) := \text{initBestand}_{\text{pnpb_k}}(\text{kIKon}_{\text{ppb_pnpb}}(\text{ppb}),k)$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $\text{initBestand}_{\text{ppb_k}}(\text{ppb},k) := \text{initBestand}_{\text{pvpb_k}}(\text{kIKon}_{\text{ppb_pvpb}}(\text{ppb}), k - \text{anzKomp}_{\text{pnpb}}(\text{kIKon}_{\text{ppb_pnpb}}(\text{ppb})))$.	363
initBestand_{pvpb,k}(pvpb,k)	<u>Funktionsbeschreibung:</u> Diese Funktion gibt den Initialwert der Komponente (k) des geplanten entwicklungsprojektspezifischen Verbrauchspotenzialbestandsvektors (pvpb) wieder. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden den entsprechenden Komponenten (k) der geplanten entwicklungsprojektspezifischen Verbrauchspotenzialvektoren (pvpb) über Annahmen im jeweiligen Entwicklungsprojekt jeweils in Form einer nicht negativen reellen Zahl zugeordnet.	353

Offener Funktionsausdruck	Bedeutung	SNr.
initBestand_{spb_k}(spb,k)	<u>Funktionsbeschreibung:</u> Diese Funktion gibt den Initialwert der Komponente (k) des simulierten entwicklungsprojektspezifischen (Wird-) Potenzialbestandsvektors (spb) wieder. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 338, 342, 354, 367 und 368 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $k \leq \text{anzKomp}_{\text{snpb}}(\text{kIKon}_{\text{spb_snpb}}(\text{spb}));$ daraus folgt: $\text{initBestand}_{\text{spb_k}}(\text{spb},k) := \text{initBestand}_{\text{snpb_k}}(\text{kIKon}_{\text{spb_snpb}}(\text{spb}),k).$ Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $\text{initBestand}_{\text{spb_k}}(\text{spb},k) := \text{initBestand}_{\text{svpb_k}}(\text{kIKon}_{\text{spb_svpb}}(\text{spb}), k - \text{anzKomp}_{\text{snpb}}(\text{kIKon}_{\text{spb_snpb}}(\text{spb}))).$	369
initBestand_{svpb_k}(svpb,k)	<u>Funktionsbeschreibung:</u> Diese Funktion gibt den Initialwert der Komponente (k) des simulierten entwicklungsprojektspezifischen (Wird-) Verbrauchspotenzialbestandsvektors (svpb) wieder. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden den entsprechenden Komponenten (k) der simulierten entwicklungsprojektspezifischen (Wird-) Verbrauchspotenzialvektoren (svpb) über Annahmen im jeweiligen Entwicklungsprojekt jeweils in Form einer nicht negativen reellen Zahl zugeordnet.	354
initBestand_{snpb_k}(snpb,k)	<u>Funktionsbeschreibung:</u> Diese Funktion gibt den Initialwert der Komponente (k) des simulierten entwicklungsprojektspezifischen (Wird-) Nutzungspotenzialbestandsvektors (snpb) wieder. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden den entsprechenden Komponenten (k) der simulierten entwicklungsprojektspezifischen (Wird-) Nutzungspotenzialvektoren (snpb) über Annahmen im jeweiligen Entwicklungsprojekt jeweils in Form einer natürlichen Zahl oder der Zahl „0“ zugeordnet.	342

Offener Funktionsausdruck	Bedeutung	SNr.
initModiOENj_pr_avl(j,pr,avl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) und dessen (Prozess-) Regel (pr) mit der Regelanwendungsvoraussetzungsliste (avl) den Wert, der für die Initialisierung der jeweiligen Konsequenz eines zugrunde liegenden Verifikationsversuchsergebnisses relevant ist. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in den in SNr. 329 bis 331 spezifizierten wff enthaltenen Inferenzregeln in Form des Modus ponens als Konsequenz in der zugehörigen Implikation zugewiesen. Dabei werden drei Fälle unterschieden. Fall 1): liegt vor, wenn die Antezedenz der Implikation der unter SNr. 329 spezifizierten wff erfüllt ist, in diesem Fall gilt: $\text{initModiOENj_pr_avl}(j,pr,avl) = 1.$ Fall 2): liegt vor, wenn die Antezedenz der Implikation der unter SNr. 330 spezifizierten wff erfüllt ist, in diesem Fall gilt: $\text{initModiOENj_pr_avl}(j,pr,avl) = 2.$ Fall 3): liegt vor, wenn die Antezedenz der Implikation der unter SNr. 331 spezifizierten wff erfüllt ist, in diesem Fall gilt: $\text{initModiOENj_pr_avl}(j,pr,avl) = 3.$	327
initZeitpunkt_{inpb}(inpb)	<u>Funktionsbeschreibung:</u> Diese Funktion gibt den Initialisierungszeitpunkt des realisierten entwicklungsprojektspezifischen (Ist-) Nutzungspotenzialbestandsvektors (inpb) wieder. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden den entsprechenden realisierten entwicklungsprojektspezifischen (Ist-) Nutzungspotenzialbestandsvektoren (inpb) über Annahmen zum entsprechenden Initialisierungszeitpunkt im jeweiligen Entwicklungsprojekt jeweils in Form einer nicht negativen reellen Zahl zugeordnet.	346
initZeitpunkt_{ipb}(ipb)	<u>Funktionsbeschreibung:</u> Diese Funktion gibt den Initialisierungszeitpunkt des realisierten entwicklungsprojektspezifischen (Ist-) Potenzialbestandsvektors (ipb) wieder. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 349 und 373 spezifizierten Funktionsausdrücke wie folgt festgelegt: $\text{initZeitpunkt}_{ipb}(ipb):$ $\text{initZeitpunkt}_{inpb}(kIKon_{ipb_inpb}(ipb),k).$	376

Offener Funktionsausdruck	Bedeutung	SNr.
initZeitpunkt_{ivpb}(ivpb)	<u>Funktionsbeschreibung:</u> Diese Funktion gibt den Initialisierungszeitpunkt des realisierten entwicklungsprojektspezifischen (Ist-) Verbrauchspotenzialbestandsvektors (ivpb) wieder. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden den entsprechenden realisierten entwicklungsprojektspezifischen (Ist-) Verbrauchspotenzialbestandsvektoren (ivpb) über Annahmen zum entsprechenden Initialisierungszeitpunkt im jeweiligen Entwicklungsprojekt jeweils in Form einer nicht negativen reellen Zahl zugeordnet.	358
initZeitpunkt_{pnpb}(pnpb)	<u>Funktionsbeschreibung:</u> Diese Funktion gibt den Initialisierungszeitpunkt des geplanten entwicklungsprojektspezifischen Nutzungspotenzialbestandsvektors (pnpb) wieder. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden den entsprechenden geplanten entwicklungsprojektspezifischen Nutzungspotenzialbestandsvektoren (pnpb) über Annahmen zum entsprechenden Initialisierungszeitpunkt im jeweiligen Entwicklungsprojekt jeweils in Form einer nicht negativen reellen Zahl zugeordnet.	344
initZeitpunkt_{ppb}(ppb)	<u>Funktionsbeschreibung:</u> Diese Funktion gibt den Initialisierungszeitpunkt des geplanten entwicklungsprojektspezifischen Potenzialbestandsvektors (ppb) wieder. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 359 und 360 spezifizierten Funktionsausdrücke wie folgt festgelegt: $\text{initZeitpunkt}_{\text{ppb}}(\text{ppb}) := \text{initZeitpunkt}_{\text{pnpb}}(\text{kIKon}_{\text{ppb_pnpb}}(\text{ppb})).$	364
initZeitpunkt_{pvpb}(pvpb)	<u>Funktionsbeschreibung:</u> Diese Funktion gibt den Initialisierungszeitpunkt des geplanten entwicklungsprojektspezifischen Verbrauchspotenzialbestandsvektors (pvpb) wieder. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden den entsprechenden geplanten entwicklungsprojektspezifischen Verbrauchspotenzialbestandsvektoren (pvpb) über Annahmen zum entsprechenden Initialisierungszeitpunkt im jeweiligen Entwicklungsprojekt jeweils in Form einer nicht negativen reellen Zahl zugeordnet.	356

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
initZeitpunkt_{snpb}(snpb)	<u>Funktionsbeschreibung</u>: Diese Funktion gibt den Initialisierungszeitpunkt des simulierten entwicklungsprojektspezifischen (Wird-) Nutzungspotenzialbestandsvektors (snpb) wieder. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden simulierten entwicklungsprojektspezifischen (Wird-) Nutzungspotenzialbestandsvektoren (snpb) über Annahmen zum entsprechenden Initialisierungszeitpunkt im jeweiligen Entwicklungsprojekt jeweils in Form einer nicht negativen reellen Zahl zugeordnet.	345
initZeitpunkt_{spb}(spb)	<u>Funktionsbeschreibung</u>: Diese Funktion gibt den Initialisierungszeitpunkt des simulierten entwicklungsprojektspezifischen (Wird-) Potenzialbestandsvektors (spb) wieder. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 348 und 367 spezifizierten Funktionsausdrücke wie folgt festgelegt: $\text{initZeitpunkt}_{\text{spb}}(\text{spb}) := \text{initZeitpunkt}_{\text{snpb}}(\text{kIKon}_{\text{spb_snpb}}(\text{spb}), \text{k}).$	370
initZeitpunkt_{svpb}(svpb)	<u>Funktionsbeschreibung</u>: Diese Funktion gibt den Initialisierungszeitpunkt des simulierten entwicklungsprojektspezifischen (Wird-) Verbrauchspotenzialbestandsvektors (svpb) wieder. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden simulierten entwicklungsprojektspezifischen (Wird-) Verbrauchspotenzialbestandsvektoren (svpb) über Annahmen zum entsprechenden Initialisierungszeitpunkt im jeweiligen Entwicklungsprojekt jeweils in Form einer nicht negativen reellen Zahl zugeordnet.	357
INPBestandV(inpb)	Etwas (inpb) ist ein realisierter entwicklungsprojektspezifischer (Ist-) Nutzungspotenzialbestandsvektor.	334
INPBV_Komp(inpb,k)	Ein realisierter entwicklungsprojektspezifischer (Ist-) Nutzungspotenzialbestandsvektor (inpb) weist die Komponente (k) auf.	339

Offener Funktionsausdruck	Bedeutung	SNr.
iOutput_j(j,ir,k=1)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines realisierten Reaktionsgrößenvektors (ir) zu dessen Komponente k=1 (d.h. der ersten Komponente); dieser Wert repräsentiert den gegebenenfalls vorliegenden realisierten Output des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten Prozesse gegebenenfalls vorliegenden Realisationsinformationen festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Istdaten über den realisierten Output (kurz <i>ro</i>) für den hier relevanten Reaktionsgrößenvektor gegeben sind; sodann gilt: iOutput_j(j,ir,k=1):= -1. Fall 2): liegt vor, wenn Istdaten für den hier relevanten Reaktionsgrößenvektor über den entsprechend realisierten Output <i>ro</i> gegeben sind, sodann gilt: iOutput_j(j,ir,k=1):= <i>ro</i>, mit $ro \in \mathbb{N}_0$.	235
iPBelegWert_j(j,agial,agia₂,k)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den realisierten Belegungswert des in seiner Liste realisierter arbeitgangspezifischer Aktionskomponenten (agial) enthaltenen, realisierten arbeitgangspezifischen Potenzialbelegungsvektors (agia₂) für dessen in der Komponente (k) angeführte Potenzialart. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der gegebenenfalls für die jeweils relevanten prozessspezifischen Arbeitsgänge vorliegenden Prozessrealisationsinformationen zugeordnet¹⁷. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn (noch) keine realisierten Belegungswerte für den hier relevanten arbeitgangspezifischen Potenzialbelegungsvektor vorhanden sind; in diesem Fall gilt: iPBelegWert_j(j,agial,agia₂,k):=0. Fall 2): liegt vor, wenn realisierte Belegungswerte für den hier relevanten arbeitgangspezifischen Potenzialbelegungsvektor vorhanden sind; sodann gilt: iPBelegWert_j(j,agial,agia₂,k):= <i>b</i>, $b \in \mathbb{N}$.	136

¹⁷ Die hier betrachteten Funktionswerte repräsentieren die Werte der Komponenten des Potenzialbelegungsvektors *agia_{j,2}* innerhalb der Spezifikation des Parameters *ap_s(up)*.

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
IPBestandV(ipb)	Etwas (ipb) ist ein realisierter entwicklungsprojektspezifischer (Ist-) Potenzialbestandsvektor.	371
IPBV_Komp(ipb,k)	Ein realisierter entwicklungsprojektspezifischer (Ist-) Potenzialbestandsvektor (ipb) weist die Komponente (k) auf.	372
iPReaktVKomp_j(j,ipr,k)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert der gegebenenfalls realisierten Potenzialreaktion, die im zugehörigen realisierten Potenzialreaktionsvektor (ipr) in dessen Komponente (k) anzugeben ist¹⁸. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden anhand der für die hier relevanten Prozesse gegebenenfalls vorliegenden Realisationsinformationen festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn zur hier relevanten Potenzialreaktion noch kein Ist-Wert (kurz <i>iw</i>) gegeben ist; sodann gilt: $\text{iPReaktVKomp}_j(j,\text{ipr},k) := 0$ Fall 2): liegt vor, wenn zur hier relevanten Potenzialreaktion ein Ist-Wert (<i>iw</i>) gegeben ist; sodann gilt: $\text{iPReaktVKomp}_j(j,\text{ipr},k) := iw, iw \in \mathbb{R}.$	247

¹⁸ Dabei wird unterstellt, dass in der Komponente (k) die realisierte Potenzialreaktion der *k*-ten Potenzialart ($k=1, \dots, m; m \in \mathbb{N}$) verwaltet wird.

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
iPVbWert_j(j,agial,agia₁,k)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den realisierten Verbrauchswert des in seiner Liste realisierter arbeitgangspezifischer Aktionskomponenten (agial) enthaltenen, realisierten arbeitgangspezifischen Potenzialverbrauchsvektors (agia₁) für dessen in der Komponente (k) angeführte Potenzialart. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der gegebenenfalls für die jeweils relevanten prozessspezifischen Arbeitsgänge vorliegenden Prozessrealisationsinformationen zugeordnet¹⁹. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn (noch) keine realisierten Verbrauchswerte für den hier relevanten arbeitgangspezifischen Potenzialverbrauchsvektor vorhanden sind; in diesem Fall gilt: $iPVbWert_j(j,agial,agia_1,k)=0$. Fall 2): liegt vor, wenn realisierte Verbrauchswerte für den hier relevanten arbeitgangspezifischen Potenzialverbrauchsvektor vorhanden sind; sodann gilt: $iPVbWert_j(j,agial,agia_1,k)=v, \quad v \in \mathbb{R}_+$.	135
IstPReaktV_Komp(ipr,k)	Ein realisierter Potenzialreaktionsvektor (ipr) weist die Komponente (k) auf.	245
IstPReaktVektor(ipr)	Etwas (ipr) ist ein realisierter Potenzialreaktionsvektor.	244
IstRV(ir)	Etwas (ir) ist ein realisierter Reaktionsgrößenvektor.	229
IstRV_Komp(ir,k)	Ein realisierter Reaktionsgrößenvektor (ir) weist die Komponente (k) auf.	231
IVPBestandV(ivpb)	Etwas (ivpb) ist ein realisierter entwicklungsprojektspezifischer (Ist-) Verbrauchspotenzialbestandsvektor.	349
IVPBV_Komp(ivpb,k)	Ein realisierter entwicklungsprojektspezifischer (Ist-) Verbrauchspotenzialbestandsvektor (ivpb) weist die Komponente (k) auf.	352

¹⁹ Die hier betrachteten Funktionswerte repräsentieren die Werte der Komponenten des Potenzialverbrauchsvektors $agia_{j,1}$ innerhalb der Spezifikation des Parameters $ap_s(up)$.

Offener Funktionsausdruck	Bedeutung	SNr.
iWerts_j(j,ir,k=4)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines realisierten Reaktionsgrößenvektors (ir) zu dessen Komponente k=4; dieser Wert repräsentiert die gegebenenfalls vorliegende realisierte Wertschöpfung des zugrunde liegenden Prozesses (bzw. die realisierte Prozessleistung). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten Prozesse gegebenenfalls vorliegenden Realisationsinformationen festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Istdaten über die realisierte Wertschöpfung (kurz rw) für den hier relevanten Reaktionsgrößenvektor gegeben sind; sodann gilt: iWerts_j(j,sr,k=4):= -1. Fall 2): liegt vor, wenn Istdaten für den hier relevanten Reaktionsgrößenvektor über die entsprechende realisierte Wertschöpfung (rw) gegeben sind, sodann gilt: iWerts_j(j,ir,k=4):= rw, mit $rw \in \mathbb{R}_+$.	238
iZielEAT_j(j,rze,k=3)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines simulierten Prozesszielerreichungsvektors (rze) zu dessen Komponente k=3 (d.h. der dritten Komponente); dieser Wert repräsentiert den gegebenenfalls vorliegenden realisierten Anfangstermin des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe des in SNr. 124 spezifizierten Funktionsausdrucks wie folgt festgelegt: iZielEAT_j(j,rze,k=3):= iATermin_j(j).	291
iZielEET_j(j,rze,k=4)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines realisierten Prozesszielerreichungsvektors (rze) zu dessen Komponente k=4 (d.h. der vierten Komponente); dieser Wert repräsentiert den gegebenenfalls vorliegenden realisierten Endtermin des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe des in SNr. 139 spezifizierten Funktionsausdrucks wie folgt festgelegt: iZielEET_j(j,rze,k=4):= iETermin_j(j).	292

Offener Funktionsausdruck	Bedeutung	SNr.
iZieleKosten_j(j,rze,k=5,ir)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines realisierten Prozesszielerreichungsvektors (rze) zu dessen Komponente k=5 (d.h. der fünften Komponente) unter Berücksichtigung des realisierten Reaktionsgrößenvektors (ir) des zugrunde liegenden Prozesses; der Funktionswert repräsentiert die gegebenenfalls vorliegenden realisierten Kosten des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe des in SNr. 237 spezifizierten Funktionsausdrucks wie folgt festgelegt: iZieleKosten_j(j,rze,k=5,ir) := iKosten_j(j,ir,k=3).	293
iZieleOutput_j(j,rze,k=1,ir)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines realisierten Prozesszielerreichungsvektors (rze) zu dessen Komponente k=1 (d.h. der ersten Komponente) unter Berücksichtigung des realisierten Reaktionsgrößenvektors (ir) des zugrunde liegenden Prozesses; der Funktionswert repräsentiert den gegebenenfalls vorliegenden realisierten Output des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe des in SNr. 235 spezifizierten Funktionsausdrucks wie folgt festgelegt: iZieleOutput_j(j,rze,k=1,ir) := iOutput_j(j,ir,k=1).	289
iZieleQual_j(j,rze,k=2)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines realisierten Prozesszielerreichungsvektors (rze) zu dessen Komponente k=2 (d.h. der zweiten Komponente); dieser Wert repräsentiert die gegebenenfalls vorliegende realisierte Qualität des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten Prozesse gegebenenfalls vorliegenden Realisationsinformationen festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Istdaten über die realisierte Qualität (kurz <i>rq</i>) für den hier relevanten realisierten Prozesszielerreichungsvektor gegeben sind; sodann gilt: iZieleQual_j(j,rze,k=2) := -1. Fall 2): liegt vor, wenn Istdaten für den hier relevanten realisierten Prozesszielerreichungsvektor über die entsprechend realisierte Qualität <i>rq</i> gegeben sind, sodann gilt: iZieleQual_j(j,rze,k=2) := <i>rq</i>, mit $rq \in [1, 6]$.	290

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
iZieleWerts_j(j,rze,k=6,ir)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines realisierten Prozesszielerreichungsvektors (rze) zu dessen Komponente k=6 (d.h. der sechsten Komponente) unter Berücksichtigung des realisierten Reaktionsgrößenvektors (ir) des zugrunde liegenden Prozesses; der Funktionswert repräsentiert die gegebenenfalls vorliegende realisierte Wertschöpfung des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe des in SNr. 238 spezifizierten Funktionsausdrucks wie folgt festgelegt: $iZieleWerts_j(j,rze,k=6,ir) := iWerts_j(j,ir,k=4)$.	294
KAusprägung(ka)	Etwas (ka) ist eine Kritizitätsausprägung.	313
kIKon_{agsi_agd}(j,agsil,agsi,agdl)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt zu einem arbeitsgangspezifischen Soll-Ist-Abweichungsgrößenvektor (agsi), der zur Liste der Soll-Ist-Abweichungsgrößenvektoren (agsil) des Prozesses (j) gehört, den korrespondierenden Arbeitsgangdatenvektor [kurz (agd)], welcher der Arbeitsgangdatenvektorliste (agdl) des betreffenden Prozesses zugeordnet ist²⁰. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Soll-Ist-Abweichungsgrößenvektors (agsi) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den Arbeitsgangdatenvektoren (agd) jener ausgewählt, der auch diese Zahl aufweist, und entsprechend als Funktionswert zugewiesen²¹.	224

²⁰ Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome $Proz_AgSoIstVListe(j,agsil)$ gem. SNr. 218, $AgSoIstVL_AgSoIstV(agsil,agsi)$ gem. SNr. 216, $Proz_AgDatVL(j,agdl)$ gem. SNr. 68 und $AgDatVL_AgDatV(agdl,agd)$ gem. SNr. 66.

²¹ Für $agsi := Agsi_i$ wird entsprechend Agd_i als Funktionswert angesetzt. Dabei wird unterstellt, dass die Arbeitsgangdatenvektoren und die arbeitsgangspezifischen Soll-Ist-Abweichungsgrößenvektoren entsprechend der Arbeitsgangfolge des hier relevanten Prozesses in aufsteigender Folge festgelegt werden.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{agsi_agia3}(j,agsil,agsi,agial)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt zu einem arbeitgangspezifischen Soll-Ist-Abweichungsgrößenvektor (agsi), der zur Liste der Soll-Ist-Abweichungsgrößenvektoren (agsil) des Prozesses (j) gehört, die korrespondierende realisierte arbeitgangspezifische Anfangsterminposition [kurz (agia₃)], die der Liste realisierter arbeitgangspezifischer Aktionskomponenten (agsil) des zugrunde liegenden Prozesses zugeordnet ist²². <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Soll-Ist-Abweichungsgrößenvektors (agsi) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den realisierten arbeitgangspezifischen Anfangsterminpositionen (agia₃) jene ausgewählt, die auch diese Zahl in der Zeichenkette nach dem Unterstrichzeichen „_“ aufweist, und entsprechend als Funktionswert zugewiesen²³.	220

²² Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome Proz_AgSoIstVListe(j,agsil) gem. SNr. 218, AgSoIstVL_AgSoIstV(agsil,agsi) gem. SNr. 216, Proz_AgIAKListe(j,agial) gem. SNr. 126 und AgIAKL_IAGATP(agial,agia₃) gem. SNr. 132.

²³ Für agsi:= Agsi₄ wird entsprechend Agia_{3_4} als Funktionswert angesetzt. Dabei wird unterstellt, dass die arbeitgangspezifischen Soll-Ist-Abweichungsgrößenvektoren und die realisierten arbeitgangspezifischen Anfangsterminpositionen entsprechend der Arbeitsgangfolge des zugrunde liegenden Prozesses in aufsteigender Folge festgelegt werden.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{agsi_agia4}(j,agsil,agsi,agial)	<u>funktionsbeschreibung</u>: Diese Funktion ermittelt zu einem arbeitgangspezifischen Soll-Ist-Abweichungsgrößenvektor (agsi), der zur Liste der Soll-Wird-Abweichungsgrößenvektoren (agsil) des Prozesses (j) gehört, die korrespondierende realisierte arbeitgangspezifische Endterminposition [kurz (agia₄)], die der Liste simulierter arbeitgangspezifischer Aktionskomponenten (agial) des zugrunde liegenden Prozesses zugeordnet ist²⁴. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Soll-Ist-Abweichungsgrößenvektors (agsi) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den realisierten arbeitgangspezifischen Endterminpositionen (agia₄) jene ausgewählt, die auch diese Zahl in der Zeichenkette nach dem Unterstrichzeichen „_“ aufweist, und entsprechend als Funktionswert zugewiesen²⁵.	222

²⁴ Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome Proz_AgSoIstVListe(j,agsil) gem. SNr. 218, AgSoIstVL_AgSoIstV(agsil,agsi) gem. SNr. 216, Proz_AgIAKLListe(j,agial) gem. SNr. 126 und AgIAKL_IaGETP(agial,agia₄) gem. SNr. 134.

²⁵ Für agsi:= Agsi₆ wird entsprechend Agsa_{4_6} als Funktionswert angesetzt. Dabei wird unterstellt, dass die arbeitgangspezifischen Soll-Ist-Abweichungsgrößenvektoren und die realisierten arbeitgangspezifischen Endterminpositionen entsprechend der Arbeitsgangfolge des zugrunde liegenden Prozesses in aufsteigender Folge festgelegt werden.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{agsi_agir}(j,agsil,agsi,agirl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem arbeitgangsspezifischen Soll-Ist-Abweichungsgrößenvektor (agsi), der zur Liste der Soll-Wird-Abweichungsgrößenvektoren (agsil) des Prozesses (j) gehört, den korrespondierenden realisierten arbeitgangsspezifischen Reaktionsgrößenvektor [kurz (agir)], welcher der Liste realisierter arbeitgangsspezifischer Reaktionsgrößenvektoren (agirl) des betreffenden Prozesses zugeordnet ist²⁶. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Soll-Ist-Abweichungsgrößenvektors (agsi) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den realisierten arbeitgangsspezifischen Reaktionsgrößenvektoren (agir) jener ausgewählt, der auch diese Zahl aufweist, und entsprechend als Funktionswert zugewiesen²⁷.	225

²⁶ Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome Proz_AgSoIstVListe(j,agsil) gem. SNr. 218, AgSoIstVL_AgSoIstV(agsil,agsi) gem. SNr. 216, Proz_AgIRVListe(j,agirl) gem. SNr. 209 und AgIstRVL_AgIstRV(agirl,agir) gem. SNr. 207.

²⁷ Für agsi:= Agsi₅ wird entsprechend Agir₅ als Funktionswert angesetzt. Dabei wird unterstellt, dass die realisierten arbeitgangsspezifischen Reaktionsgrößenvektoren und die arbeitgangsspezifischen Soll-Ist-Abweichungsgrößenvektoren entsprechend der Arbeitsgangfolge des hier relevanten Prozesses in aufsteigender Folge festgelegt werden.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{agsi_agp}(j,agsil,agsi,agpt)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem arbeitsgangspezifischen Soll-Ist-Abweichungsgrößenvektor (agsi), der zur Liste der Soll-Ist-Abweichungsgrößenvektoren (agsil) des Prozesses (j) gehört, den korrespondierenden Arbeitsplanterminvektor [kurz (agp)], welcher der Arbeitsgangplanterminvektorenliste (agpt) des betreffenden Prozesses zugeordnet ist²⁸. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Soll-Ist-Abweichungsgrößenvektors (agsi) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den Arbeitsplanterminvektoren (agp) jener ausgewählt, der auch diese Zahl aufweist, und entsprechend als Funktionswert zugewiesen²⁹.	219
kIKon_{agsw_agd}(j,agswl,agsw,agdl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem arbeitsgangspezifischen Soll-Wird-Abweichungsgrößenvektor (agsw), der zur Liste der Soll-Wird-Abweichungsgrößenvektoren (agswl) des Prozesses (j) gehört, den korrespondierenden Arbeitsgangdatenvektor [kurz (agd)], welcher der Arbeitsgangdatenvektorenliste (agdl) des betreffenden Prozesses zugeordnet ist³⁰. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Soll-Wird-Abweichungsgrößenvektors (agsw) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den Arbeitsgangdatenvektoren (agd) jener ausgewählt, der auch diese Zahl aufweist, und entsprechend als Funktionswert zugewiesen³¹.	159

²⁸ Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome $\text{Proz_AgSoIstVListe}(j, \text{agsil})$ gem. SNr. 218, $\text{AgSoIstVL_AgSoIstV}(\text{agsil}, \text{agsi})$ gem. SNr. 216, $\text{Proz_AgPTermVL}(j, \text{agpt})$ gem. SNr. 104 und $\text{AgTVL_AgPTermV}(\text{agpt}, \text{agp})$ gem. SNr. 102.

²⁹ Für $\text{agsi} := \text{Agsi}_8$ wird entsprechend Agp_8 als Funktionswert angesetzt. Dabei wird unterstellt, dass die Arbeitsgangplanterminvektoren und die arbeitsgangspezifischen Soll-Ist-Abweichungsgrößenvektoren entsprechend der Arbeitsgangfolge des hier relevanten Prozesses in aufsteigender Folge festgelegt werden.

³⁰ Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome $\text{Proz_AgSoWiVListe}(j, \text{agswl})$ gem. SNr. 153, $\text{AgSoWiVL_AgSoWiV}(\text{agswl}, \text{agsw})$ gem. SNr. 151, $\text{Proz_AgDatVL}(j, \text{agdl})$ gem. SNr. 68 und $\text{AgDatVL_AgDatV}(\text{agdl}, \text{agd})$ gem. SNr. 66.

³¹ Für $\text{agsw} := \text{Ags}_2$ wird entsprechend Agd_2 als Funktionswert angesetzt. Dabei wird unterstellt, dass die Arbeitsgangdatenvektoren und die arbeitsgangspezifischen Soll-Wird-Abweichungsgrößenvektoren entsprechend der Arbeitsgangfolge des hier relevanten Prozesses in aufsteigender Folge festgelegt werden.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{agsw_agp}(j,agswl,agsw,agpt)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem arbeitsgangspezifischen Soll-Wird-Abweichungsgrößenvektor (agsw), der zur Liste der Soll-Wird-Abweichungsgrößenvektoren (agswl) des Prozesses (j) gehört, den korrespondierenden Arbeitsplanterminvektor [kurz (agp)], welcher der Arbeitsgangplanterminvektorenliste (agpt) des betreffenden Prozesses zugeordnet ist³². <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Soll-Wird-Abweichungsgrößenvektors (agsw) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den Arbeitsplanterminvektoren (agp) jener ausgewählt, der auch diese Zahl aufweist, und entsprechend als Funktionswert zugewiesen³³.	154

³² Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome `Proz_AgSoWiVListe(j,agswl)` gemäß SNr. 153, `AgSoWiVL_AgSoWiV(agswl,agsw)` gem. SNr. 151, `Proz_AgPTermVL(j,agpt)` gemäß SNr. 104 und `AgTVL_AgPTermV(agpt,agp)` gem. SNr. 102.

³³ Für `agsw:=Agsw9` wird entsprechend `Apg9` als Funktionswert angesetzt. Dabei wird unterstellt, dass die Arbeitsgangplanterminvektoren und die arbeitsgangspezifischen Soll-Wird-Abweichungsgrößenvektoren entsprechend der Arbeitsgangfolge des hier relevanten Prozesses in aufsteigender Folge festgelegt werden.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{agsw_agsa3}(j,agswl,agsw,agsal)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem arbeitgangspezifischen Soll-Wird-Abweichungsgrößenvektor (agsw), der zur Liste der Soll-Wird-Abweichungsgrößenvektoren (agswl) des Prozesses (j) gehört, die korrespondierende simulierte arbeitgangspezifische Anfangsterminposition [kurz (agsa₃)], die der Liste simulierter arbeitgangspezifischer Aktionskomponenten (agsal) des zugrunde liegenden Prozesses zugeordnet ist³⁴. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Soll-Wird-Abweichungsgrößenvektors (agsw) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den simulierten arbeitgangspezifischen Anfangsterminpositionen (agsa₃) jene ausgewählt, die auch diese Zahl in der Zeichenkette nach dem Unterstrichzeichen „_“ aufweist, und entsprechend als Funktionswert zugewiesen³⁵.	155

³⁴ Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome `Proz_AgSoWiVListe(j,agswl)` gem. SNr. 153, `AgSoWiV_AgSoWiV(agswl,agsw)` gem. SNr. 151, `Proz_AgSAKListe(j,agsal)` gem. SNr. 110 und `AgSAKL_SAgATP(agsal,agsa3)` gem. SNr. 116.

³⁵ Für `agsw:=Agsw9` wird entsprechend `Agsa3_9` als Funktionswert angesetzt. Dabei wird unterstellt, dass die arbeitgangspezifischen Soll-Wird-Abweichungsgrößenvektoren und die simulierten arbeitgangspezifischen Anfangsterminpositionen entsprechend der Arbeitsgangfolge des zugrunde liegenden Prozesses in aufsteigender Folge festgelegt werden.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{agsw_agsa4}(j,agswl,agsw,agsal)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem arbeitgangspezifischen Soll-Wird-Abweichungsgrößenvektor (agsw), der zur Liste der Soll-Wird-Abweichungsgrößenvektoren (agswl) des Prozesses (j) gehört, die korrespondierende simulierte arbeitgangspezifische Endterminposition [kurz (agsa₄)], die der Liste simulierter arbeitgangspezifischer Aktionskomponenten (agsal) des zugrunde liegenden Prozesses zugeordnet ist³⁶. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Soll-Wird-Abweichungsgrößenvektors (agsw) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den simulierten arbeitgangspezifischen Endterminpositionen (agsa₄) jene ausgewählt, die auch diese Zahl in der Zeichenkette nach dem Unterstrichzeichen „_“ aufweist, und entsprechend als Funktionswert zugewiesen³⁷.	157

³⁶ Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome Proz_AgSoWiVListe(j,agswl) gem. SNr. 153, AgSoWiVL_AgSoWiV(agswl,agsw) gem. SNr. 151, Proz_AgSAKListe(j,agsal) gem. SNr. 110 und AgSAKL_SAgETP(agsal,agsa₄) gem. SNr. 118.

³⁷ Für agsw:= Agsw₇ wird entsprechend Agsa_{4_7} als Funktionswert angesetzt. Dabei wird unterstellt, dass die arbeitgangspezifischen Soll-Wird-Abweichungsgrößenvektoren und die simulierten arbeitgangspezifischen Endterminpositionen entsprechend der Arbeitsgangfolge des zugrunde liegenden Prozesses in aufsteigender Folge festgelegt werden.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{agsw_agr}(j,agswl,agsw,agsrl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem arbeitsgangspezifischen Soll-Wird-Abweichungsgrößenvektor (agsw), der zur Liste der Soll-Wird-Abweichungsgrößenvektoren (agswl) des Prozesses (j) gehört, den korrespondierenden simulierten arbeitsgangspezifischen Reaktionsgrößenvektor [kurz (agsr)], welcher der Liste simulierter arbeitsgangspezifischer Reaktionsgrößenvektoren (agsrl) des betreffenden Prozesses zugeordnet ist³⁸. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Soll-Wird-Abweichungsgrößenvektors (agsw) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den simulierten arbeitsgangspezifischen Reaktionsgrößenvektoren (agsr) jener ausgewählt, der auch diese Zahl aufweist, und entsprechend als Funktionswert zugewiesen³⁹.	160

³⁸ Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome `Proz_AgSoWiVListe(j,agswl)` gem. SNr. 153, `AgSoWiV_AgSoWiV(agswl,agsw)` gem. SNr. 151, `Proz_AgSRVListe(j,agsrl)` gem. SNr. 144 und `AgSimRVL_AgSimRV(agsrl,agsr)` gem. SNr. 142.

³⁹ Für `agsw:=Agsw3` wird entsprechend `Agsw3` als Funktionswert angesetzt. Dabei wird unterstellt, dass die simulierten arbeitsgangspezifischen Reaktionsgrößenvektoren und die arbeitsgangspezifischen Soll-Wird-Abweichungsgrößenvektoren entsprechend der Arbeitsgangfolge des hier relevanten Prozesses in aufsteigender Folge festgelegt werden.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{agwi_agia3}(j,agwil,agwi,agial)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektor (agwi), der zur Liste der Wird-Ist-Abweichungsgrößenvektoren (agwil) des Prozesses (j) gehört, die korrespondierende realisierte arbeitsgangspezifische Anfangsterminposition [kurz (agia₃)], die der Liste realisierter arbeitsgangspezifischer Aktionskomponenten (agsil) des zugrunde liegenden Prozesses zugeordnet ist⁴⁰. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Wird-Ist-Abweichungsgrößenvektors (agwi) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den realisierten arbeitsgangspezifischen Anfangsterminpositionen (agia₃) jene ausgewählt, die auch diese Zahl in der Zeichenkette nach dem Unterstrichzeichen „_“ aufweist, und entsprechend als Funktionswert zugewiesen⁴¹.	258

⁴⁰ Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome Proz_AgWiIstVListe(j,agwil) gem. SNr. 256, AgWiIstVL_AgWiIst(agwil,agwi) gem. SNr. 254, Proz_AgIAKListe(j,agial) gem. SNr. 126 und AgIAKL_IAGATP(agial,agia₃) gem. SNr. 132.

⁴¹ Für agwi:= Agwi₂ wird entsprechend Agia_{3_2} als Funktionswert angesetzt. Dabei wird unterstellt, dass die arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektoren und die realisierten arbeitsgangspezifischen Anfangsterminpositionen entsprechend der Arbeitsgangfolge des zugrunde liegenden Prozesses in aufsteigender Folge festgelegt werden.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{agwi_agia4}(j,agwil,agwi, agial)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt zu einem arbeitgangspezifischen Soll-Ist-Abweichungsgrößenvektor (agwi), der zur Liste der Wird-Ist-Abweichungsgrößenvektoren (agwil) des Prozesses (j) gehört, die korrespondierende realisierte arbeitgangspezifische Endterminposition [kurz (agia₄)], die der Liste simulierter arbeitgangspezifischer Aktionskomponenten (agial) des zugrunde liegenden Prozesses zugeordnet ist⁴². <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Wird-Ist-Abweichungsgrößenvektors (agwi) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den realisierten arbeitgangspezifischen Endterminpositionen (agia₄) jene ausgewählt, die auch diese Zahl in der Zeichenkette nach dem Unterstrichzeichen „_“ aufweist, und entsprechend als Funktionswert zugewiesen⁴³.	261

⁴² Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome $\text{Proz_AgWiIstVListe}(j, \text{agwil})$ gem. SNr. 256, $\text{AgWiIstVL_AgWiIst}(\text{agwil}, \text{agwi})$ gem. SNr. 254, $\text{Proz_AgIAKListe}(j, \text{agial})$ gem. SNr. 126 und $\text{AgIAKL_IAgETP}(\text{agial}, \text{agia}_4)$ gem. SNr. 134.

⁴³ Für $\text{agwi} := \text{Agwi}_5$ wird entsprechend Agia_{4_5} als Funktionswert angesetzt. Dabei wird unterstellt, dass die arbeitgangspezifischen Wird-Ist-Abweichungsgrößenvektoren und die realisierten arbeitgangspezifischen Endterminpositionen entsprechend der Arbeitsgangfolge des zugrunde liegenden Prozesses in aufsteigender Folge festgelegt werden.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{agwi_agir}(j,agwil,agwi,agirl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektor (agwi), der zur Liste der Wird-Ist-Abweichungsgrößenvektoren (agwil) des Prozesses (j) gehört, den korrespondierenden realisierten arbeitsgangspezifischen Reaktionsgrößenvektor [kurz (agir)], welcher der Liste realisierter arbeitsgangspezifischer Reaktionsgrößenvektoren (agirl) des betreffenden Prozesses zugeordnet ist⁴⁴. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Wird-Ist-Abweichungsgrößenvektors (agwi) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den realisierten arbeitsgangspezifischen Reaktionsgrößenvektoren (agir) jener ausgewählt, der auch diese Zahl aufweist, und entsprechend als Funktionswert zugewiesen⁴⁵.	264

⁴⁴ Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome $\text{Proz_AgWiIstVListe}(j, \text{agwil})$ gem. SNr. 256, $\text{AgWiIstVL_AgWiIst}(\text{agwil}, \text{agwi})$ gem. SNr. 254, $\text{Proz_AgIRVListe}(j, \text{agirl})$ gem. SNr. 209 und $\text{AgIstRVL_AgIstRV}(\text{agirl}, \text{agir})$ gem. SNr. 207.

⁴⁵ Für $\text{agwi} := \text{Agwi}_i$ wird entsprechend Agir_i als Funktionswert angesetzt. Dabei wird unterstellt, dass die realisierten arbeitsgangspezifischen Reaktionsgrößenvektoren und die arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektoren entsprechend der Arbeitsgangfolge des hier relevanten Prozesses in aufsteigender Folge festgelegt werden.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{agwi_agsa3}(j,agwil,agwi,agsal)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektor (agwi), der zur Liste der Wird-Ist-Abweichungsgrößenvektoren (agwil) des Prozesses (j) gehört, die korrespondierende simulierte arbeitsgangspezifische Anfangsterminposition [kurz (agsa₃)], die der Liste simulierter arbeitsgangspezifischer Aktionskomponenten (agsal) des zugrunde liegenden Prozesses zugeordnet ist⁴⁶. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Wird-Ist-Abweichungsgrößenvektors (agwi) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den simulierten arbeitsgangspezifischen Anfangsterminpositionen (agsa₃) jene ausgewählt, die auch diese Zahl in der Zeichenkette nach dem Unterstrichzeichen „_“ aufweist, und entsprechend als Funktionswert zugewiesen⁴⁷.	257

⁴⁶ Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome Proz_AgWiIstVListe(j,agwil) gem. SNr. 256, AgWiIstVL_AgWiIst(agwil,agwi) gem. SNr. 254, Proz_AgSAKLListe(j,agsal) gem. SNr. 110 und AgSAKL_SAgATP(agsal,agsa₃) gem. SNr. 116.

⁴⁷ Für agwi:= Agwi₄ wird entsprechend Agsa_{3_4} als Funktionswert angesetzt. Dabei wird unterstellt, dass die arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektoren und die simulierten arbeitsgangspezifischen Anfangsterminpositionen entsprechend der Arbeitsgangfolge des zugrunde liegenden Prozesses in aufsteigender Folge festgelegt werden.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{agwi_agsa4}(j,agwil,agwi,agsal)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektor (agwi), der zur Liste der Wird-Ist-Abweichungsgrößenvektoren (agwil) des Prozesses (j) gehört, die korrespondierende simulierte arbeitsgangspezifische Endterminposition [kurz (agsa₄)], die der Liste simulierter arbeitsgangspezifischer Aktionskomponenten (agsal) des zugrunde liegenden Prozesses zugeordnet ist⁴⁸. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Wird-Ist-Abweichungsgrößenvektors (agwi) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den simulierten arbeitsgangspezifischen Endterminpositionen (agsa₄) jene ausgewählt, die auch diese Zahl in der Zeichenkette nach dem Unterstrichzeichen „_“ aufweist, und entsprechend als Funktionswert zugewiesen⁴⁹.	260

⁴⁸ Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome Proz_AgWiIstVListe(j,agwil) gem. SNr. 256, AgWiIstVL_AgWiIst(agwil,agwi) gem. SNr. 254, Proz_AgSAKLListe(j,agsal) gem. SNr. 110 und AgSAKL_SAgETP(agsal,agsa₄) gem. SNr. 118.

⁴⁹ Für agwi:= Agwi₁ wird entsprechend Agsa_{4_1} als Funktionswert angesetzt. Dabei wird unterstellt, dass die arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektoren und die simulierten arbeitsgangspezifischen Endterminpositionen entsprechend der Arbeitsgangfolge des zugrunde liegenden Prozesses in aufsteigender Folge festgelegt werden.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{agwi_agsr}(j,agwil,agwi,agsrl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektor (agwi), der zur Liste der Wird-Ist-Abweichungsgrößenvektoren (agwil) des Prozesses (j) gehört, den korrespondierenden simulierten arbeitsgangspezifischen Reaktionsgrößenvektor [kurz (agsr)], welcher der Liste simulierter arbeitsgangspezifischer Reaktionsgrößenvektoren (agsrl) des betreffenden Prozesses zugeordnet ist⁵⁰. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des Wird-Ist-Abweichungsgrößenvektors (agwi) enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den simulierten arbeitsgangspezifischen Reaktionsgrößenvektoren (agsr) jener ausgewählt, der auch diese Zahl aufweist, und entsprechend als Funktionswert zugewiesen⁵¹.	263
kIKon_{ipb_inpb}(ipb)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem realisierten entwicklungsprojektspezifischen (Ist-) Potenzialbestandsvektor (ipb) den korrespondierenden entwicklungsprojektspezifischen realisierten (Ist-) Nutzungspotenzialbestandsvektor (inpb). <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des realisierten entwicklungsprojektspezifischen (Ist-) Potenzialbestandsvektors enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den realisierten entwicklungsprojektspezifischen (Ist-) Nutzungspotenzialvektoren (inpb) jener ausgewählt, der auch diese Zahl aufweist, und entsprechend als Funktionswert zugewiesen⁵².	373

⁵⁰ Die hier erwähnten Korrespondenzen lassen sich zurückführen auf die Atome Proz_AgWiIstVListe(j,agwil) gem. SNr. 256, AgWiIstVL_AgWiIst(agwil,agwi) gem. SNr. 254, Proz_AgSRVListe(j,agsrl) gem. SNr. 144 und AgSimRVL_AgSimRV(agsrl,agsr) gem. SNr. 142.

⁵¹ Für agwi:= Agwi₃ wird entsprechend Agsr₃ als Funktionswert angesetzt. Dabei wird unterstellt, dass die simulierten arbeitsgangspezifischen Reaktionsgrößenvektoren und die arbeitsgangspezifischen Wird-Ist-Abweichungsgrößenvektoren entsprechend der Arbeitsgangfolge des hier relevanten Prozesses in aufsteigender Folge festgelegt werden.

⁵² Für ipb:= Ipb₂ wird entsprechend Inpb₂ als Funktionswert angesetzt.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{ipb_ivpb}(ipb)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem realisierten entwicklungsprojekt-spezifischen (Ist-) Potenzialbestandsvektor (ipb) den korrespondierenden entwicklungsprojekt-spezifischen realisierten (Ist-) Verbrauchspotenzialbestandsvektor (inpb). <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des realisierten entwicklungsprojektspezifischen (Ist-) Potenzialbestandsvektors enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den realisierten entwicklungsprojektspezifischen (Ist-) Verbrauchspotenzialvektoren (ivpb) jener ausgewählt, der auch diese Zahl aufweist, und entsprechend als Funktionswert zugewiesen⁵³.	374
kIKon_{ppb_pnpb}(ppb)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem geplanten entwicklungsprojekt-spezifischen Potenzialbestandsvektor (ppb) den korrespondierenden entwicklungsprojektspezifischen geplanten Nutzungspotenzialbestandsvektor (pnpb). <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des geplanten entwicklungsprojektspezifischen Potenzialbestandsvektors enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den geplanten entwicklungsprojektspezifischen Nutzungspotenzialvektoren (pnpb) jener ausgewählt, der auch diese Zahl aufweist, und entsprechend als Funktionswert zugewiesen⁵⁴.	360

⁵³ Für ipb:= Ipb₂ wird entsprechend Inpb₂ als Funktionswert angesetzt.

⁵⁴ Für ppb:= Ppb₉ wird entsprechend Pnpb₉ als Funktionswert angesetzt.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{ppb_pvpb}(ppb)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem geplanten entwicklungsprojektspezifischen Potenzialbestandsvektor (ppb) den korrespondierenden entwicklungsprojektspezifischen geplanten Verbrauchspotenzialbestandsvektor (pvpb). <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des geplanten entwicklungsprojektspezifischen Potenzialbestandsvektors enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den geplanten entwicklungsprojektspezifischen Verbrauchspotenzialvektoren (pvpb) jener ausgewählt, der auch diese Zahl aufweist, und entsprechend als Funktionswert zugewiesen⁵⁵.	361
kIKon_{spb_snpb}(spb)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem simulierten entwicklungsprojektspezifischen (Wird-) Potenzialbestandsvektor (spb) den korrespondierenden entwicklungsprojektspezifischen simulierten (Wird-) Nutzungspotenzialbestandsvektor (snpb). <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des simulierten entwicklungsprojektspezifischen (Wird-) Potenzialbestandsvektors enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den simulierten entwicklungsprojektspezifischen (Wird-) Nutzungspotenzialvektoren (snpb) jener ausgewählt, der auch diese Zahl aufweist, und entsprechend als Funktionswert zugewiesen⁵⁶.	367

⁵⁵ Für ppb:= Ppb₇ wird entsprechend Pvpb₇ als Funktionswert angesetzt.

⁵⁶ Für spb:= Spb₂ wird entsprechend Snpb₂ als Funktionswert angesetzt.

Offener Funktionsausdruck	Bedeutung	SNr.
kIKon_{spb_svpb}(spb)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt zu einem simulierten entwicklungsprojekt-spezifischen (Wird-) Potenzialbestandsvektor (spb) den korrespondierenden entwicklungsprojektspezifischen simulierten (Wird-) Verbrauchspotenzialbestandsvektor (svpb). <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden wie folgt ermittelt: Entsprechend der in der jeweiligen Individuenkonstanten des simulierten entwicklungsprojektspezifischen (Wird-) Potenzialbestandsvektors enthaltenen Zahl wird als korrespondierende Individuenkonstante aus den simulierten entwicklungsprojektspezifischen (Wird-) Verbrauchspotenzialvektoren (svpb) jener ausgewählt, der auch diese Zahl aufweist, und entsprechend als Funktionswert zugewiesen⁵⁷.	368
kritAnzNegVeri_a(j,pr,avl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) die kritische Anzahl der zur (Prozess-) Regel (pr) mit der Regelanwendungsvoraussetzungsliste (avl) tolerierten unzureichenden Verifikationsversuchsergebnisse. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden über Annahmen im jeweiligen Projektentwicklungskontext in Form einer natürlichen Zahl direkt zugewiesen. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn der Prozess (j) kein BS-Prozess ist, in diesem Fall gilt: $\text{kritAnzNegVeri_a}(j,pr,avl) := 0$. Fall 2): liegt vor, wenn der Prozess (j) ein BS-Prozess ist, in diesem Fall gilt: $\text{kritAnzNegVeri_a}(j,pr,avl) := n_{knv}$, mit $n_{knv} \in \mathbb{N}_{>0}$.	324
minBedG(cs)	<u>Funktionsbeschreibung:</u> Diese Funktion gibt zu dem Controlling-Zeitpunkt, welcher der Controlling-Zeitstufe (cs) zugeordnet ist, den dann zu berücksichtigenden Mindestwert für die bedingten Eintrittsgewichte der zu analysierenden (Unter-) Prozesse wieder. <u>Funktionsgleichung:</u> Der jeweilige Funktionswert wird mithilfe der in SNr. 381 und 382 spezifizierten Funktionsausdrücke wie folgt ermittelt: $\text{minBedG}(cs) := (\text{dMinLokG}(cs))^{\text{anzEProzMinLokG}(cs)}.$	383

⁵⁷ Für spb:= Spb₄ wird entsprechend Svpb₄ als Funktionswert angesetzt.

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
mZeitpunkt(ms)	<u>Funktionsbeschreibung</u>: Diese Funktion gibt den der Metacontrolling-Zeitstufe (ms) zuzuordnenden Metacontrolling-Zeitpunkt wieder. <u>Funktionsgleichung</u>: Ausgehend von den im zugrunde liegenden Entwicklungsprojekt von den projektverantwortlichen Institutionen festgelegten Metacontrolling-Zeitpunkten $tm_{j_{tm}} \in \mathbb{R}_+$ (mit $j_{tm} = 1, \dots, J_{tm} \in \mathbb{N}_{>0}$) wird jeweils der Metacontrolling-Zeitpunkt der Metacontrolling-Zeitstufe (ms) zugeordnet für dessen Indexwert gilt: $j_{tm} = ms$, d.h.: $mZeitpunkt(ms) := tp_{ms}$.	380
MZeitstufe(ms)	Etwas (ms) ist eine Metacontrolling-Zeitstufe.	378
NachfolgerProzNr(npn)	Etwas (npn) ist eine Nachfolgerprozessnummer.	35
NachProzNrListe(npnl)	Etwas (npnl) ist eine Nachfolgerprozessnummernliste.	30
NPBedarfVektor(np)	Etwas (np) ist ein Nutzungspotenzialbedarfsvektor.	79
NPBedV_Komp(pn,k)	Ein Nutzungspotenzialbedarfsvektor (np) weist die Komponente (k) auf.	80
npBedarfVKomp(j,np,k)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des Bedarfs an Nutzungspotenzial, das im zugehörigen (prozessspezifischen) Nutzungspotenzialbedarfsvektor (np) in dessen Komponente (k) angeführt ist. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt über Annahmen für die relevanten Prozesse in Form einer natürlichen Zahl einschließlich der Zahl „0“ zugeordnet.	82
NPNListe_NProzNr(npnl,npn)	Eine Nachfolgerprozessnummernliste (npnl) beinhaltet die Nachfolgerprozessnummer (npn).	36
pAgKostenj(j,agdl,agd,k=8)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Arbeitsgangdatenvektorenliste (agdl) enthaltenen Arbeitsgangdatenvektors (agd) zu dessen Komponente k=8; dieser Wert repräsentiert die Plankosten des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt über Annahmen für die relevanten Prozesse in Form einer nicht negativen reellen Zahl zugeordnet.	76

Offener Funktionsausdruck	Bedeutung	SNr.
pAgOutput_j(j,agdl,agd,k=7)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Arbeitsgangdatenvektorenliste (agdl) enthaltenen Arbeitsgangdatenvektors (agd) zu dessen Komponente k=7; dieser Wert stellt den Planarbeitsgangoutput des betreffenden Arbeitsgangs dar. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt über Annahmen für die relevanten Prozesse in Form einer natürlichen Zahl einschließlich der Zahl 0 zugeordnet.	75
pAgWerts_j(j,agdl,agd,k=9)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Arbeitsgangdatenvektorenliste (agdl) enthaltenen Arbeitsgangdatenvektors (agd) zu dessen Komponente k=9; der Wert repräsentiert die Planwertschöpfung des entsprechenden Arbeitsgangs. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt über Annahmen für die relevanten Prozesse in Form einer nicht negativen reellen Zahl zugeordnet.	77
pATermin_j(j)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt den Plananfangstermin des Prozesses (j). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden Prozessen (j) anhand der im relevanten (Teil-) Projektkontext verfügbaren Planungsinformationen direkt zugeordnet; dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn noch kein Plananfangstermin für den Prozess (j) gegeben ist; in diesem Fall gilt: $pATermin_j(j) := -1$. Fall 2): liegt vor, wenn ein Plananfangstermin für den Prozess (j) gegeben ist; sodann gilt: $pATermin_j(j) := t, \quad t \in \mathbb{R}_+.$	99

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
pbwert_{j,pr,bppzl,bp,pz}(j,pr,bppzl,bp,pz)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt den geplanten Wert der Beitragsposition (bp), die dem Projektziel (pz) zugeordnet ist und in der Liste Projektzielen zugeordneter Beitragspositionen (bppzl) enthalten ist, welche für die Prozessregel (pr) spezifiziert ist, die im Prozess (j) zur Anwendung vorgesehen ist. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden über entsprechende Annahmen für die Anwendung der Prozessregel (pr) im Prozess (j) im jeweiligen Projektkontext zugeordnet. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, falls die Individuenkonstante „None“ für (pz) gegeben ist, sodann gilt: $\text{pbwert}_{j,pr,bppzl}(j,pr,bppzl,bp,pz) := \text{nd}$ („nd“ steht hier für „nicht definiert“); Fall 2) liegt vor, falls Fall 1) nicht zutrifft; sodann wird als Wert, je nach Projektziel, entweder eine reelle Zahl oder ein alphanumerischer Ausdruck zugewiesen.	25
pETermin_j(j)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt den Planendtermin des Prozesses (j). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden Prozessen (j) anhand der im relevanten (Teil-) Projektkontext verfügbaren Planungsinformationen direkt zugeordnet. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn noch kein Planendtermin für den Prozess (j) vorliegt; in diesem Fall gilt: $\text{pETermin}_j(j) := -1$. Fall 2): liegt vor, wenn ein Plananfangstermin für den Prozess (j) vorliegt; sodann gilt: $\text{pETermin}_j(j) := t$, mit $t \in \mathbb{R}_+$ und $\text{pETermin}_j(j) \geq \text{pATermin}_j(j)$.	107
pKosten_j(j)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt die Plankosten des Prozesses (j). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden Prozessen (j) über Annahmen im relevanten (Teil-) Projektkontext jeweils in Form einer nicht negativen reellen Zahl direkt zugeordnet.	61

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
PlanPReaktVektor(ppr)	Etwas (ppr) ist ein Planpotenzialreaktionsvektor.	87
PMerkmal_Prozessart(j,pt)	Ein Prozess (j) weist das Merkmal (pt) auf.	3
PNPBestandV(pnpb)	Etwas (pnpb) ist ein geplanter entwicklungsprojektspezifischer Nutzungspotenzialbestandsvektor.	332
PNPBV_Komp(pnpb,k)	Ein geplanter entwicklungsprojektspezifischer Nutzungspotenzialbestandsvektor (pnpb) weist die Komponente (k) auf.	335
pOutput_j(j)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt den Planoutput des Prozesses (j). <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden den entsprechenden Prozessen (j) über Annahmen im relevanten (Teil-) Projektkontext jeweils in Form einer natürlichen Zahl einschließlich der Zahl „0“ direkt zugeordnet.	60
PPBestandV(ppb)	Etwas (ppb) ist ein geplanter entwicklungsprojektspezifischer Potenzialbestandsvektor.	359
PPBV_Komp(ppb,k)	Ein geplanter entwicklungsprojektspezifischer Potenzialbestandsvektor (ppb) weist die Komponente (k) auf.	362
PPReaktV_Komp(ppr,k)	Ein Planpotenzialreaktionsvektor (ppr) weist die Komponente (k) auf.	88
pPReaktVKomp_j(j,ppr,k)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert der geplanten Potenzialreaktion, die im zugehörigen (prozessspezifischen) Planpotenzialreaktionsvektor (ppr) in dessen Komponente (k) anzugeben ist ⁵⁸ . <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden direkt über Annahmen für die relevanten Prozesse in Form einer reellen Zahl zugeordnet.	90
PRegel(pr)	Etwas (pr) ist eine (Prozess-) Regel.	14
Projektziel(pz)	Etwas (pz) ist ein Projektziel.	19
Prozess(j)	Etwas (j) ist ein (Unter-) Prozess.	1
Prozessart(pt)	Etwas (pt) ist eine Prozessart.	2
Prozess_SIListe(j,sil)	Ein Prozess (j) besitzt die Steuerungsinstitutionsliste (sil).	13
Proz_AgDatVL(j,agdl)	Ein Prozess (j) besitzt eine Arbeitsgangdatenvektorenliste (agdl).	68

⁵⁸ Dabei wird unterstellt, dass in der Komponente (k) die geplante Potenzialreaktion der k -ten Potenzialart ($k=1, \dots, m$; $m \in \mathbb{N}$) verwaltet wird.

Offenes Atom	Bedeutung	SNr.
Proz_AgIAKListe(j, agial)	Ein Prozess (j) besitzt eine Liste realisierter arbeitgangspezifischer Aktionskomponenten (agial).	126
Proz_AgIRVListe(j, agirl)	Ein Prozess (j) besitzt eine Liste (agirl) realisierter arbeitgangspezifischer Reaktionsgrößenvektoren.	209
Proz_AgPTermVL(j, agpt)	Ein Prozess (j) besitzt eine Arbeitsgangplanterminkomponentenliste (agpt).	104
Proz_AgSAKListe(j, agsal)	Ein Prozess (j) besitzt eine Liste simulierter arbeitgangspezifischer Aktionskomponenten (agsal).	110
Proz_AgSoIstVListe(j, agsil)	Ein Prozess (j) besitzt eine Liste (agsil) arbeitgangspezifischer Soll-Ist-Abweichungsgrößenvektoren.	218
Proz_AgSoWiVListe(j, agswl)	Ein Prozess (j) besitzt eine Liste (agswl) arbeitgangspezifischer Soll-Wird-Abweichungsgrößenvektoren.	153
Proz_AgSRVListe(j, agsrl)	Ein Prozess (j) besitzt eine Liste (agsrl) simulierter arbeitgangspezifischer Reaktionsgrößenvektoren.	144
Proz_AgWiIstVListe(j, agwil)	Ein Prozess (j) besitzt eine Liste (agwil) arbeitgangspezifischer Wird-Ist-Abweichungsgrößenvektoren.	256
Proz_IPReaktV(j, ipr)	Ein Prozess (j) besitzt einen realisierten Potenzialreaktionsvektor (ipr).	246
Proz_IstRV(j, ir)	Ein Prozess (j) besitzt einen realisierten Reaktionsgrößenvektor (ir).	230
Proz_NPBedV(j, np)	Ein Prozess (j) besitzt einen Nutzungspotenzialbedarfsvektor (np).	81
Proz_NPNListe(j, npnl)	Ein Prozess (j) besitzt eine Nachfolgerprozessnummernliste (vpnl).	37
Proz_PPReaktV(j, ppr)	Ein Prozess (j) besitzt den Planpotenzialreaktionsvektor (ppr).	89
Proz_PR_AVL(j, pr, avl)	Ein Prozess (j) besitzt zu seiner (Prozess-) Regel (pr) die Regelanwendungsvoraussetzungsliste (avl).	314
Proz_PR_BPPZL(j, pr, bppzl)	Ein Prozess (j) besitzt zu seiner Prozessregel (pr) eine Liste (bppzl) von Beitragspositionen, die Projektzielen zugeordnet sind.	24
Proz_Pregel(j, pr)	Ein Prozess (j) besitzt eine (Prozess-) Regel (pr).	15
Proz_Prüfregel(j, cr)	Ein Prozess (j) besitzt eine Prüfregel (cr).	318
Proz_PZZL (j, przl)	Ein Prozess (j) besitzt die Prozesszielliste (przl).	50

Offenes Atom	Bedeutung	SNr.
Proz_RAZL(j,lat)	Ein Prozess (j) besitzt eine Liste relevanter Aktionszeitpunktpositionen (lat).	54
Proz_RProzZieleV(j,rze)	Ein Prozess (j) besitzt einen realisierten Prozesszielerreichungsvektor (rze).	288
Proz_SimRV(j,sr)	Ein Prozess (j) besitzt einen simulierten Reaktionsgrößenvektor (sr).	165
Proz_SoIstPRV(j,sipr)	Ein Prozess (j) besitzt einen Soll-Ist-Potenzialreaktionsabweichungsvektor (sipr).	250
Proz_SoIstV(j,sia)	Ein Prozess (j) besitzt einen Soll-Ist-Abweichungsgrößenvektor (sia).	233
Proz_SoIstZV(j,siz)	Ein Prozess (j) besitzt einen Soll-Ist-Zielabweichungsvektor (siz).	297
Proz_SoWiPRV(j,swpr)	Ein Prozess (j) besitzt einen Soll-Wird-Potenzialreaktionsabweichungsvektor (swpr).	185
Proz_SoWiV(j,swa)	Ein Prozess (j) besitzt einen Soll-Wird-Abweichungsgrößenvektor (swa).	168
Proz_SoWiZV(j,swz)	Ein Prozess (j) besitzt einen Soll-Wird-Zielabweichungsvektor (swz).	198
Proz_SPReaktV(j,spr)	Ein Prozess (j) besitzt einen simulierten prozessspezifischen Potenzialreaktionsvektor (spr).	181
Proz_SProzZieleV(j,sze)	Ein Prozess (j) besitzt einen simulierten Prozesszielerreichungsvektor (sze).	188
Proz_UZL (j,uzl)	Ein Prozess (j) besitzt eine Liste (uzl) von Umweltzuständen.	46
Proz_VPBedV(j,vp)	Ein Prozess (j) besitzt den Verbrauchspotenzialvektor (vp).	85
Proz_VPNListe(j,vpnl)	Ein Prozess (j) besitzt eine Vorgängerprozessnummernliste (vpnl).	33
Proz_WiIstPRV(j,wipr)	Ein Prozess (j) besitzt einen Wird-Ist-Potenzialreaktionsabweichungsvektor (wipr).	273
Proz_WiIstV(j,wia)	Ein Prozess (j) besitzt einen Wird-Ist-Abweichungsgrößenvektor (wia).	269
Proz_WiIstZV(j,wiz)	Ein Prozess (j) besitzt einen Wird-Ist-Zielabweichungsvektor (wiz).	300
ProzZiel(prz)	Etwas (prz) ist ein Prozessziel.	48
ProzZielListe(przl)	Etwas (przl) ist eine Prozesszielliste.	47
Prüfregel(cr)	Etwas (cr) ist eine Prüfregel.	317
PVPBestandV(pvpb)	Etwas (pvpb) ist ein geplanter entwicklungsprojektspezifischer Verbrauchspotenzialbestandsvektor.	347
PVPBV_Komp(pvpb,k)	Ein geplanter entwicklungsprojektspezifischer Verbrauchspotenzialbestandsvektor (pvpb) weist die Komponente (k) auf.	350

Offener Funktionsausdruck	Bedeutung	SNr.
pWerts_j(j)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt die Planwertschöpfung des Prozesses (j). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden Prozessen (j) über Annahmen im relevanten (Teil-) Projektkontext jeweils in Form einer nicht negativen reellen Zahl direkt zugeordnet.	62
pzielwert(pz)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt den Wert (der Ausprägung) des Projektziels (pz). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden über entsprechende Annahmen im jeweiligen Projekt den Projektzielen (pz) zugeordnet. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, falls für (pz) die Individuenkonstante „None“ gegeben ist, sodann gilt: pzielwert(None):=nd, („nd“ steht hier für „nicht definiert“); Fall 2) liegt vor, falls Fall 1) nicht zutrifft; sodann wird als Wert, je nach Projektziel, entweder eine reelle Zahl oder ein alphanumerischer Ausdruck zugewiesen.	20
pzSkalaKNr(pz)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt die Kennnummer, welche zur Kennzeichnung der dem Prozessziel (pz) zugrunde liegenden Skalierung dient. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden über entsprechende Annahmen bzgl. der Operationalisierung des Prozessziels (pz) im jeweiligen Projektkontext direkt zugeordnet. Dabei werden vier Fälle unterschieden. Fall 1): liegt vor, sofern für das Projektziel (pz) die Individuenkonstante „None“ gegeben ist, sodann gilt: pzSkalaKNr(pz):= 4 (als Kennzeichen für nicht operationalisiert). Fall 2): liegt vor, sofern Fall 1 nicht vorliegt und das zugrunde liegende Projektziel nominal skaliert ist, sodann gilt: pzSkalaKNr(pz):= 3. Fall 3): liegt vor, sofern Fall 1 nicht vorliegt und das zugrunde liegende Projektziel ordinal skaliert ist, sodann gilt: pzSkalaKNr(pz):= 2. Fall 4): liegt vor, sofern Fall 1 nicht vorliegt und das zugrunde liegende Projektziel kardinal skaliert ist, sodann gilt: pzSkalaKNr(pz):= 1.	280

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
PZZListe_ProzZiel(przl, prz)	Eine Prozesszielliste (przl) beinhaltet ein Prozessziel (prz).	49
RAktZeitListe(lat)	Etwas (lat) ist eine Liste relevanter Aktionszeitpunktpositionen.	51
RAktZeitPos(at)	Etwas (at) ist eine relevante Aktionszeitpunktposition.	52
RAZListe_RAZPos(lat, at)	Eine Liste relevanter Aktionszeitpunktpositionen (lat) beinhaltet die Aktionszeitpunktposition (at).	53
RProzZieleVektor(rze)	Etwas (rze) ist ein realisierter Prozesszielerreichungsvektor.	286
RProzZieleV_Komp(rze,k)	Ein realisierter Prozesszielerreichungsvektor (rze) weist die Komponente (k) auf.	287
sAgAT_j(j,agsal, agsa₃)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den simulierten Arbeitsganganfangstermin zu der in seiner Liste der simulierten arbeitsgangspezifischen Aktionskomponenten (agsal) enthaltenen simulierten Arbeitsganganfangsterminposition (agsa₃). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der gegebenenfalls für die relevanten prozessspezifischen Arbeitsgänge vorliegenden Simulationslaufinformationen zugeordnet⁵⁹. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn (noch) kein simulierter Anfangstermin für den hier relevanten prozessspezifischen Arbeitsgang vorhanden ist; in diesem Fall gilt: $sAgAT_j(j,agsal,agsa_3) := -1.$ Fall 2): liegt vor, wenn ein simulierter Anfangstermin für den hier relevanten prozessspezifischen Arbeitsgang gegeben ist; sodann gilt: $sAgAT_j(j,agsal,agsa_3) := ts, \quad ts \in \mathbb{R}_+.$	121
SAgATPos(agsa₃)	Etwas (agsa ₃) ist eine simulierte arbeitsgangspezifische Anfangsterminposition.	115

⁵⁹ Die hier betrachteten Funktionswerte repräsentieren jeweils den simulierten Arbeitsganganfangstermin des Arbeitsgangs j , $agsa_{j,3}$, innerhalb der Spezifikation des Parameters $ap_2(up)$.

Offener Funktionsausdruck	Bedeutung	SNr.
sAgAusschuss_j(j,agsrl,agsr,k=2)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste simulierter arbeitgangspezifischer Reaktionsgrößenvektoren (agsrl) enthaltenen simulierten arbeitgangspezifischen Reaktionsgrößenvektors (agsr) zu dessen Komponente k=2; dieser Wert repräsentiert den gegebenenfalls vorliegenden simulierten Ausschuss des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten prozessspezifischen Arbeitsgänge gegebenenfalls vorliegenden Simulationsdaten festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch kein echter simulierter arbeitgangspezifischer Reaktionsgrößenvektor in der betreffenden Liste (agsrl) vorhanden ist [(agsr) ist dabei mit der Individuenkonstante „None“ belegt] oder noch keine Simulationsdaten über den simulierten Ausschuss (kurz <i>sa</i>) für den hier relevanten Reaktionsgrößenvektor gegeben sind; sodann gilt: $\text{sAgAusschuss}_j(j,\text{agsrl},\text{agsr},k=2) := -1.$ Fall 2): liegt vor, wenn Simulationsdaten für den hier relevanten arbeitgangspezifischen Reaktionsgrößenvektor über den entsprechend simulierten Ausschuss <i>sa</i> vorhanden sind, sodann gilt: $\text{sAgAusschuss}_j(j,\text{agsrl},\text{agsr},k=2) := sa,$ mit $\text{agsr} \in \text{AGSR} \setminus \{\text{None}\}, sa \in \mathbb{N}_0.$	146

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
sAgET_j(j,agsal,agsa₄)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den simulierten Arbeitsgangendtermin zu der in seiner Liste der simulierten arbeitsgangsspezifischen Aktionskomponenten (agsal) enthaltenen simulierten Arbeitsgangendterminposition (agsa₄). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der gegebenenfalls für die relevanten prozessspezifischen Arbeitsgänge vorliegenden Simulationslaufinformationen zugeordnet⁶⁰. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn (noch) kein simulierter Endtermin für den hier relevanten prozessspezifischen Arbeitsgang vorhanden ist; in diesem Fall gilt: $\text{sAgET}_j(j, \text{agsal}, \text{agsa}_4) := -1.$ Fall 2): liegt vor, wenn ein simulierter Endtermin für den hier relevanten prozessspezifischen Arbeitsgang vorhanden ist; sodann gilt: $\text{sAgET}_j(j, \text{agsal}, \text{agsa}_4) := ts, \quad ts \in \mathbb{R}_+.$	122
SAgETPos(agsa₄)	Etwas (agsa ₄) ist eine simulierte arbeitsgangsspezifische Endterminposition.	117

⁶⁰ Die hier betrachteten Funktionswerte repräsentieren jeweils den simulierten Arbeitsgangendtermin des Arbeitsgangs j , $\text{agsa}_{j,4}$, innerhalb der Spezifikation des Parameters $ap_2(up)$.

Offener Funktionsausdruck	Bedeutung	SNr.
sAgKosten_j(j,agsrl,agsr,k=3)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste simulierter arbeitgangspezifischer Reaktionsgrößenvektoren (agsrl) enthaltenen simulierten arbeitgangspezifischen Reaktionsgrößenvektors (agsr) zu dessen Komponente k=3; dieser Wert repräsentiert die gegebenenfalls vorliegenden simulierten Kosten des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten prozessspezifischen Arbeitsgänge gegebenenfalls vorliegenden Simulationsdaten festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch kein echter simulierter arbeitgangspezifischer Reaktionsgrößenvektor in der betreffenden Liste (agsrl) vorhanden ist [(agsr) ist dabei mit der Individuenkonstante „None“ belegt] oder noch keine Simulationsdaten über die simulierten Kosten (kurz <i>sk</i>) für den hier relevanten Reaktionsgrößenvektor gegeben sind; in diesem Fall gilt: $\text{sAgKosten}_j(j, \text{agsrl}, \text{agsr}, k=3) := -1.$ Fall 2): liegt vor, wenn Simulationsdaten für den hier relevanten arbeitgangspezifischen Reaktionsgrößenvektor über die entsprechend simulierten Kosten <i>sk</i> vorhanden sind, sodann gilt: $\text{sAgKosten}_j(j, \text{agsrl}, \text{agsr}, k=3) := sk,$ mit $\text{agsr} \in \text{AGSR} \setminus \{\text{None}\}$, $sk \in \mathbb{R}_+$.	147

Offener Funktionsausdruck	Bedeutung	SNr.
sAgOutput_j(j,agsrl,agsr,k=1)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste simulierter arbeitgangspezifischer Reaktionsgrößenvektoren (agsrl) enthaltenen, simulierten arbeitgangspezifischen Reaktionsgrößenvektors (agsr) zu dessen Komponente k=1 (d.h. der ersten Komponente); dieser Wert repräsentiert den gegebenenfalls vorliegenden simulierten Output des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten prozessspezifischen Arbeitsgänge gegebenenfalls vorliegenden Simulationsdaten festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch kein echter simulierter arbeitgangspezifischer Reaktionsgrößenvektor in der betreffenden Liste (agsrl) vorhanden ist [(agsr) ist dabei mit der Individuenkonstante „None“ belegt] oder noch keine Simulationsdaten über den simulierten Output (kurz <i>so</i>) für den hier relevanten Reaktionsgrößenvektor gegeben sind; sodann gilt: $\text{sAgOutput}_j(j, \text{agsrl}, \text{agsr}, k=1) := -1.$ Fall 2): liegt vor, wenn Simulationsdaten für den hier relevanten arbeitgangspezifischen Reaktionsgrößenvektor über den entsprechend simulierten Output <i>so</i> vorhanden sind, sodann gilt: $\text{sAgOutput}_j(j, \text{agsrl}, \text{agsr}, k=1) := so,$ mit $\text{agsr} \in \text{AGSR} \setminus \{\text{None}\}, so \in \mathbb{N}_0.$	145

Offener Funktionsausdruck	Bedeutung	SNr.
sAgPBelegWert_j(j,agsal,agsa₂,k)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den simulierten Belegungswert des in seiner Liste simulierter arbeitgangspezifischer Aktionskomponenten (agsal) enthaltenen simulierten arbeitgangspezifischen Potenzialbelegungsvektors (agsa₂) für dessen in der Komponente (k) angeführte Potenzialart. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der gegebenenfalls für die relevanten prozessspezifischen Arbeitsgänge vorliegenden Simulationslaufinformationen zugeordnet⁶¹. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn (noch) keine simulierten Belegungswerte für den hier relevanten arbeitgangspezifischen Potenzialbelegungsvektor vorhanden sind; in diesem Fall gilt: sAgPBelegWert_j(j,agsal,agsa₂,k):=0. Fall 2): liegt vor, wenn simulierte Belegungswerte für den hier relevanten arbeitgangspezifischen Potenzialbelegungsvektor vorhanden sind; sodann gilt: sAgPBelegWert_j(j,agsal,agsa₂,k):= b, b ∈ ℕ.	120

⁶¹ Die hier betrachteten Funktionswerte repräsentieren die Werte der Komponenten des Potenzialbelegungsvektors $agsa_{j,2}$ innerhalb der Spezifikation des Parameters $ap_2(up)$.

Offener Funktionsausdruck	Bedeutung	SNr.
sAgPVbWert_j(j,agsal,agsa₁,k)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den simulierten Verbrauchswert des in seiner Liste simulierter arbeitgangspezifischer Aktionskomponenten (agsal) enthaltenen simulierten arbeitgangspezifischen Potenzialverbrauchsvektors (agsa₁) für dessen in der Komponente (k) angeführte Potenzialart. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der gegebenenfalls für die relevanten prozessspezifischen Arbeitsgänge vorliegenden Simulationslaufinformationen zugeordnet⁶². Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn (noch) keine simulierten Verbrauchswerte für den hier relevanten arbeitgangspezifischen Potenzialverbrauchsvektor vorhanden sind; in diesem Fall gilt: $\text{sAgPVbWert}_j(j,agsal,agsa_1,k) := 0$. Fall 2): liegt vor, wenn simulierte Verbrauchswerte für den hier relevanten arbeitgangspezifischen Potenzialverbrauchsvektor vorhanden sind; sodann gilt: $\text{sAgPVbWert}_j(j,agsal,agsa_1,k) := v, \quad v \in \mathbb{R}_+$.	119

⁶² Die hier betrachteten Funktionswerte repräsentieren die Werte der Komponenten des Potenzialverbrauchsvektors $agsa_{j,1}$ innerhalb der Spezifikation des Parameters $ap_2(up)$.

Offener Funktionsausdruck	Bedeutung	SNr.
sAgWSchöpf_j(j,agsrl,agsr,k=4)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des in seiner Liste simulierter arbeitgangspezifischer Reaktionsgrößenvektoren (agsrl) enthaltenen simulierten arbeitgangspezifischen Reaktionsgrößenvektors (agsr) zu dessen Komponente k=4; dieser Wert repräsentiert die gegebenenfalls vorliegende simulierte Wertschöpfung des zugrunde liegenden Arbeitsgangs. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten prozessspezifischen Arbeitsgänge gegebenenfalls vorliegenden Simulationsdaten festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch kein echter simulierter arbeitgangspezifischer Reaktionsgrößenvektor in der betreffenden Liste (agsrl) vorhanden ist [(agsr) ist dabei mit der Individuenkonstante „None“ belegt] oder noch keine Simulationsdaten über die simulierte Wertschöpfung (kurz <i>sw</i>) für den hier relevanten Reaktionsgrößenvektor gegeben sind; sodann gilt: $\text{sAgWSchöpf}_j(j, \text{agsrl}, \text{agsr}, k=4) := -1.$ Fall 2): liegt vor, wenn Simulationsdaten für den hier relevanten arbeitgangspezifischen Reaktionsgrößenvektor über die entsprechend simulierte Wertschöpfung <i>sw</i> vorhanden sind, sodann gilt: $\text{sAgWSchöpf}_j(j, \text{agsrl}, \text{agsr}, k=4) := sw,$ mit $\text{agsr} \in \text{AGSR} \setminus \{\text{None}\}$, $sw \in \mathbb{R}_+$.	148

Offener Funktionsausdruck	Bedeutung	SNr.
sATermin_j(j)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt den simulierten Anfangstermin des Prozesses (j). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden Prozessen (j) anhand der im relevanten (Teil-) Projektkontext im aktuellen bzw. letzten Simulationslauf verfügbaren Simulationsdaten direkt zugeordnet⁶³; dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn (noch) kein simulierter Anfangstermin für den Prozess (j) gegeben ist; sodann gilt: $sATermin_j(j) := -1$. Fall 2): liegt vor, wenn ein simulierter Anfangstermin für den Prozess (j) gegeben ist; sodann gilt: $sATermin_j(j) := ts$, $ts \in \mathbb{R}_+$.	108
sAusschuss_j(j,sr,k=2)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines simulierten Reaktionsgrößenvektors (sr) zu dessen Komponente k=2; dieser Wert repräsentiert den gegebenenfalls vorliegenden simulierten Ausschuss des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten Prozesse gegebenenfalls vorliegenden Simulationsdaten festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Simulationsdaten über den simulierten Ausschuss (kurz <i>sa</i>) für den hier relevanten Reaktionsgrößenvektor gegeben sind; sodann gilt: $sAusschuss_j(j,sr,k=2) := -1$. Fall 2): liegt vor, wenn Simulationsdaten für den hier relevanten Reaktionsgrößenvektor über den entsprechend simulierten Ausschuss <i>sa</i> gegeben sind, sodann gilt: $sAusschuss_j(j,sr,k=2) := sa$, mit $sa \in \mathbb{N}_0$.	171

⁶³ Im Simulationskontext wird hinsichtlich der Zuordnung der Funktionswerte unterstellt, dass beim Eintritt des simulierten Startereignisses des Prozesses (j) im aktuellen Simulationslauf der zugehörige Ereigniszeitpunkt (d.h. der simulierte Startzeitpunkt des Prozesses (j)) der Funktion sATermin_j(j) als Funktionswert zugeordnet wird.

Offener Funktionsausdruck	Bedeutung	SNr.
sbwert_{j_pr_bppzl}(j,pr,bppzl,bp,pz)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt den simulierten Wert der Beitragsposition (bp), die dem Projektziel (pz) zugeordnet ist und in der Liste Projektzielen zugeordneter Beitragspositionen (bppzl) enthalten ist, welche für die Prozessregel (pr) spezifiziert ist, die im Prozess (j) zur Anwendung vorgesehen ist. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden über entsprechende Annahmen für die Anwendung der Prozessregel (pr) im Prozess (j) im jeweiligen Projektkontext zugeordnet. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, falls die Individuenkonstante „None“ für das Projektziel (pz) gegeben ist, sodann gilt: $\text{sbwert}_{j_pr_bppzl}(j,pr,bppzl,bp,pz) := nd \text{ („nd“ steht hier für „nicht definiert“).}$ Fall 2) liegt vor, falls Fall 1) nicht zutrifft und noch keine Simulationsdaten über den Wert der zugrunde liegenden Beitragsposition gegeben sind; sodann gilt: $\text{sbwert}_{j_pr_bppzl}(j,pr,bppzl,bp,pz) := nv \text{ („nv“ steht hier für „no value“ bzw. „kein Wert“).}$ Fall 3) liegt vor, falls Fall 1) nicht zutrifft und Simulationsdaten über den Wert der zugrunde liegenden Beitragsposition gegeben sind; sodann wird als Wert, je nach Projektziel, entweder eine reelle Zahl oder ein alphanumerischer Ausdruck zugewiesen.	281
sETermin_j(j)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt den simulierten Endtermin des Prozesses (j). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden Prozessen (j) anhand der im relevanten (Teil-) Projektkontext im aktuellen bzw. letzten Simulationslauf verfügbaren Simulationsdaten direkt zugeordnet⁶⁴. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn (noch) kein simulierter Endtermin für den Prozess (j) vorhanden ist; sodann gilt: $\text{sETermin}_j(j) := -1$. Fall 2): liegt vor, wenn ein simulierter Endtermin für den Prozess (j) vorhanden ist; sodann gilt: $\text{sETermin}_j(j) := ts, \quad ts \in \mathbb{R}_+$.	123

⁶⁴ Im Simulationskontext wird hinsichtlich der Zuordnung der Funktionswerte unterstellt, dass beim Eintritt des simulierten Endereignisses des Prozesses (j) im aktuellen Simulationslauf der zugehörige Ereigniszeitpunkt $\text{sETermin}_j(j)$ als Funktionswert zugeordnet wird.

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
SimPReaktVektor(spr)	Etwas (spr) ist ein simulierter Potenzialreaktionsvektor.	179
SimPReaktV_Komp(spr,k)	Ein simulierter Potenzialreaktionsvektor (spr) weist die Komponente (k) auf.	180
SimRV(sr)	Etwas (sr) ist ein simulierter Reaktionsgrößenvektor.	164
SimRV_Komp(sr,k)	Ein simulierter Reaktionsgrößenvektor (sr) weist die Komponente (k) auf.	166
SInstitution(si)	Etwas (si) ist eine Steuerungsinstitution.	10
SInstListe(sil)	Etwas (sil) ist eine Steuerungsinstitutionsliste.	11
SIListe_SInst(sil,si)	Eine Steuerungsinstitutionsliste (sil) beinhaltet die Steuerungsinstitution (si).	12
sKosten_j(j,sr,k=3)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines simulierten Reaktionsgrößenvektors (sr) zu dessen Komponente k=3; dieser Wert repräsentiert die gegebenenfalls vorliegenden simulierten Kosten des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten Prozesse gegebenenfalls vorliegenden Simulationsdaten festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Simulationsdaten über die simulierten Kosten (kurz sk) für den hier relevanten Reaktionsgrößenvektor gegeben sind; sodann gilt: $sKosten_j(j,sr,k=3) := -1$. Fall 2): liegt vor, wenn Simulationsdaten für den hier relevanten Reaktionsgrößenvektor über die entsprechenden simulierten Kosten sk gegeben sind, sodann gilt: $sKosten_j(j,sr,k=3) := sk$, mit $sk \in \mathbb{R}_+$.	172
SNPBestandV(snpb)	Etwas (snpb) ist ein simulierter entwicklungsprojektspezifischer (Wird-) Nutzungspotenzialbestandsvektor.	333
SNPBV_Komp(snpb,k)	Ein simulierter entwicklungsprojektspezifischer (Wird-) Nutzungspotenzialbestandsvektor (snpb) weist die Komponente (k) auf.	337

Offener Funktionsausdruck	Bedeutung	SNr.
soistATA_j(j,sia,k=1)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Ist-Abweichungsgrößenvektors (sia) zu dessen Komponente k=1 (d.h. der ersten Komponente); dieser Wert repräsentiert die Soll-Ist-Anfangs-terminabweichung des zugrunde liegenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 99 und 124 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $pATermin_j(j) = -1$ (d.h., es liegt noch kein Plananfangstermin für den hier relevanten Prozess vor) oder wenn gilt: $iATermin_j(j) = -1$ (d.h., es liegt noch kein realisierter Anfangstermin für den hier relevanten Prozess vor), dann folgt daraus: $soistATA_j(j,sia,k=1) := 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $soistATA_j(j,sia,k=1) := pATermin_j(j) - iATermin_j(j)$.	239
soistETA_j(j,sia,k=2)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Ist-Abweichungsgrößenvektors (sia) zu dessen Komponente k=2; dieser Wert repräsentiert die Soll-Ist-Endterminabweichung des zugrunde liegenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 107 und 139 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $pETermin_j(j) = -1$ (d.h., es liegt noch kein Planendtermin für den hier relevanten Prozess vor) oder wenn gilt: $iETermin_j(j) = -1$ (d.h., es liegt noch kein realisierter Endtermin für den hier relevanten Prozess vor), dann folgt daraus: $soistETA_j(j,sia,k=2) := 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $soistETA_j(j,sia,k=2) := pETermin_j(j) - iETermin_j(j)$.	240

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
soistKA_j(j,sia,k=4,ir)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Ist-Abweichungsgrößenvektors (sia) zu dessen Komponente k=4 unter Berücksichtigung des zugehörigen realisierten prozessspezifischen Reaktionsgrößenvektors (ir); dieser Wert repräsentiert die Soll-Ist-Kostenabweichung des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 61 und 237 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $iKosten_j(j,ir,k=3) = -1$ (d.h., es liegen noch keine Istdaten über die realisierten Kosten für den hier relevanten Prozess vor), dann folgt daraus: $soistKA_j(j,sia,k=4,ir) := 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $soistKA_j(j,sia,k=4,ir) := pKosten_j(j) - iKosten_j(j,ir,k=3)$.	242
soistOpA_j(j,sia,k=3,ir)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Ist-Abweichungsgrößenvektors (sia) zu dessen Komponente k=3 unter Berücksichtigung des zugehörigen realisierten prozessspezifischen Reaktionsgrößenvektors (ir); dieser Wert repräsentiert die Soll-Ist-Outputabweichung des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 60 und 235 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $iOutput_j(j,ir,k=1) = -1$ (d.h., es liegen noch keine Istdaten über den realisierten Output für den hier relevanten Prozess vor), dann folgt daraus: $soistOpA_j(j,sia,k=3,ir) := 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $soistOpA_j(j,sia,k=3,ir) := pOutput_j(j) - iOutput_j(j,ir,k=1)$.	241
SoIstPRV_Komp(sipr,k)	Ein Soll-Ist-Potenzialreaktionsabweichungsvektor (sipr) weist die Komponente (k) auf.	249

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
soistPRVKomp_j(j,sipr,k,ppr,ipr)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Ist-Potenzialreaktionsabweichungsvektors (sipr) zu dessen Komponente (k) unter Berücksichtigung des Wertes der geplanten Potenzialreaktion, die im Planpotenzialreaktionsvektor (ppr) des hier zugrunde liegenden Prozesses in der übereinstimmenden Komponente (k) als Sollgröße angegeben ist, sowie des Wertes der realisierten Potenzialreaktion, die im realisierten Potenzialreaktionsvektor (ipr) des hier relevanten Prozesses in der wiederum übereinstimmenden Komponente (k) als Istgröße angegeben ist⁶⁵. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 90 und 247 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn zu der hier relevanten Istgröße noch keine Simulationsdaten vorhanden sind; sodann gilt: soistPRVKomp_j(j,sipr,k,ppr,ipr):= 0. Fall 2): liegt vor, wenn zu der hier relevanten Potenzialreaktion Istdaten gegeben sind; sodann gilt: soistPRVKomp_j(j,sipr,k,ppr,ipr):= pPReaktVKomp_j(j,ppr,k) – iPReaktVKomp_j(j,ipr,k).	251
SoIstPRVektor(sipr)	Etwas (sipr) ist ein Soll-Ist-Potenzialreaktionsabweichungsvektor.	248
SoIstV(sia)	Etwas (sia) ist ein Soll-Ist-Abweichungsgrößenvektor.	232
SoIstV_Komp(sia,k)	Ein Soll-Ist-Abweichungsgrößenvektor (sia) weist die Komponente (k) auf.	234

⁶⁵ Dabei wird unterstellt, dass in der Komponente (k) die Soll-Ist-Potenzialreaktionsabweichung der k -ten Potenzialart ($k=1, \dots, m$; $m \in \mathbb{N}_{>0}$) verwaltet wird.

Offener Funktionsausdruck	Bedeutung	SNr.
soistawert_{j_pr_bppzl}(j,pr,bppzl, bp,pz)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt den Soll-Ist-Abweichungswert bezüglich der Beitragsposition (bp), die dem Projektziel (pz) zugeordnet ist und in der Liste Projektzielen zugeordneter Beitragspositionen (bppzl) enthalten ist, welche für die Prozessregel (pr) spezifiziert ist, die im Prozess (j) zur Anwendung vorgesehen ist. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 25, 280 und 283 spezifizierten Funktionsausdrücke festgelegt. Dabei werden drei Fälle unterschieden. Fall 1): liegt vor, falls die Individuenkonstante „None“ für das Projektziel (pz) gegeben ist oder der zugehörige Funktionswert von pzSkalaKNr(pz) größer als „1“ ist, sodann gilt: $\text{soistawert}_{j_pr_bppzl}(j,pr,bppzl,bp,pz) := nd \quad (,nd“ \text{ steht hier für „nicht definiert“}).$ Fall 2): liegt vor, falls Fall 1) nicht zutrifft und für das jeweils relevante Projektziel (pz) der Funktionswert von pzSkalaKNr(pz) „1“ beträgt und noch keine Simulationsdaten oder noch keine Realisationsdaten über den Wert der zugrunde liegenden Beitragsposition gegeben sind; sodann gilt: $\text{soistawert}_{j_pr_bppzl}(j,pr,bppzl,bp,pz) := nv \quad (,nv“ \text{ steht hier für „no value“ bzw. „kein Wert“}).$ Fall 3): liegt vor, falls Fall 1) nicht zutrifft und für das jeweils relevante Projektziel (pz) der Funktionswert von pzSkalaKNr(pz) „1“ beträgt und sowohl Simulationsdaten als auch Realisationsdaten über den Wert der zugrunde liegenden Beitragsposition gegeben sind; sodann gilt: $\text{soistawert}_{j_pr_bppzl}(j,pr,bppzl,bp,pz) := \text{pbwert}_{j_pr_bppzl}(j,pr,bppzl,bp,pz) - \text{ibwert}_{j_pr_bppzl}(j,pr,bppzl,bp,pz).$	284

Offener Funktionsausdruck	Bedeutung	SNr.
soistWsA_j(j,sia,k=5,ir)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Ist-Abweichungsgrößenvektors (sia) zu dessen Komponente k=5 unter Berücksichtigung des zugehörigen realisierten prozessspezifischen Reaktionsgrößenvektors (ir); dieser Wert repräsentiert die Soll-Ist-Wertschöpfungsabweichung des zugrunde liegenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 62 und 238 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $iWerts_j(j,ir,k=4) = -1$ (d.h., es liegen noch keine Istdaten über die realisierte Wertschöpfung für den hier relevanten Prozess vor), dann folgt daraus: $soistWsA_j(j,sia,k=5,ir) := 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $soistWsA_j(j,sia,k=5,ir) := pWerts_j(j) - iWerts_j(j,ir,k=4)$.	243

Offener Funktionsausdruck	Bedeutung	SNr.
soistZielATA_j(j,siz,k=3,przl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Ist-Zielabweichungsvektors (siz) zu dessen Komponente k=3 (der dritten Komponente) unter Berücksichtigung der Prozesszielliste (przl) des zugrunde liegenden Prozesses; der Funktionswert repräsentiert die Soll-Ist-Anfangsterminzielabweichung des relevanten Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 55 und 124 spezifizierten Funktionsausdrücke festgelegt. Dabei werden drei Fälle unterschieden⁶⁶. Fall 1): liegt vor, wenn noch keine Daten über den realisierten Anfangstermin des jeweiligen Prozesses gegeben sind oder Daten über den realisierten Anfangstermin gegeben sind und folgende Bedingung erfüllt ist: $\text{wert}_{j_przl_prz}(j,przl,Prz_3) \leq iATermin_j(j) \leq \text{wert}_{j_przl_prz}(j,przl,Prz_4),$ dann liegt keine (signifikante) Terminabweichung vor, dann folgt: $\text{soistZielATA}_j(j,siz,k=3,przl) := 0$ Fall 2): liegt vor, falls Daten über den realisierten Anfangstermin gegeben sind und folgende Bedingung erfüllt ist: $iATermin_j(j) < \text{wert}_{j_przl_prz}(j,przl,Prz_3), \quad \text{dann}$ liegt eine Terminverfrühung vor und es folgt: $\text{soistZielATA}_j(j,siz,k=3,przl) := \text{wert}_{j_przl_prz}(j,przl,Prz_3) - iATermin_j(j)$ Fall 3): liegt vor, falls Daten über den realisierten Anfangstermin gegeben sind und folgende Bedingung erfüllt ist: $iATermin_j(j) > \text{wert}_{j_przl_prz}(j,przl,Prz_4), \quad \text{dann}$ liegt eine Terminverspätung vor und es folgt: $\text{soistZielATA}_j(j,siz,k=3,przl) := \text{wert}_{j_przl_prz}(j,przl,Prz_4) - iATermin_j(j).$	303

⁶⁶ Wie bei SNr. 52 angemerkt, repräsentieren im Folgenden die Individuenkonstante „Prz₃“ das Ziel des frühesten Anfangstermins und die Individuenkonstante „Prz₄“ das Ziel des spätesten Anfangstermins (entsprechend den Prozesszielen pcz_3 und pcz_4 innerhalb der Spezifikation von Parameter $dp_6(up)$).

Offener Funktionsausdruck	Bedeutung	SNr.
soistZielETA_j(j,siz,k=4,przl)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Ist-Zielabweichungsvektors (siz) zu dessen Komponente k=4 (der vierten Komponente) unter Berücksichtigung der Prozesszielliste (przl) des zugrunde liegenden Prozesses; der Funktionswert repräsentiert die Soll-Ist-Endterminzielabweichung des betreffenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 55 und 139 spezifizierten Funktionsausdrücke festgelegt. Dabei werden drei Fälle unterschieden⁶⁷. Fall 1): liegt vor, wenn noch keine Daten über den realisierten Endtermin des jeweiligen Prozesses gegeben sind oder Daten über den realisierten Endtermin gegeben sind und folgende Bedingung erfüllt ist: $\text{wert}_{j_przl_prz}(j,przl,Prz_5) \leq iETermin_j(j) \leq \text{wert}_{j_przl_prz}(j,przl,Prz_6),$ dann liegt keine (signifikante) Terminabweichung vor, dann folgt: $\text{soistZielETA}_j(j,siz,k=4,przl) := 0$ Fall 2): liegt vor, falls Daten über den realisierten Endtermin gegeben sind und folgende Bedingung erfüllt ist: $iETermin_j(j) < \text{wert}_{j_przl_prz}(j,przl,Prz_5), \quad \text{dann}$ liegt eine Terminverfrühung vor und es folgt: $\text{soistZielETA}_j(j,siz,k=4,przl) := \text{wert}_{j_przl_prz}(j,przl,Prz_5) - iETermin_j(j)$ Fall 3): liegt vor, falls Daten über den realisierten Endtermin gegeben sind und folgende Bedingung erfüllt ist: $iETermin_j(j) > \text{wert}_{j_przl_prz}(j,przl,Prz_6), \quad \text{dann}$ liegt eine Terminverspätung vor und es folgt: $\text{soistZielETA}_j(j,siz,k=4,przl) := \text{wert}_{j_przl_prz}(j,przl,Prz_6) - iETermin_j(j).$	304

⁶⁷ Wie bei SNr. 48 angemerkt, repräsentiert im Folgenden die Individuenkonstante „Prz₅“ das Ziel des frühesten Endtermins und die Individuenkonstante „Prz₆“ das Ziel des spätesten Endtermins (entsprechend den Prozesszielen pcz_5 und pcz_6 innerhalb der Spezifikation von Parameter $dp_6(up)$).

Offener Funktionsausdruck	Bedeutung	SNr.
soistZielKostenA_j(j,siz, k=5,przl,ir)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Ist-Zielabweichungsvektors (siz) zu dessen Komponente k=5 (der fünften Komponente) unter Berücksichtigung der Prozesszielliste (przl) des zugrunde liegenden Prozesses und dessen realisierten Reaktionsgrößenvektors (ir); der Funktionswert repräsentiert die Soll-Ist-Kostenzielabweichung des hier relevanten Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 55 und 237 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Daten über die realisierten Kosten des zugrunde liegenden Prozesses gegeben sind; sodann gilt: $\text{soistZielKostenA}_j(j, \text{siz}, k=5, \text{przl}, \text{ir}) = 0$. Fall 2): liegt vor, wenn Daten über die realisierten Kosten des zugrunde liegenden Prozesses gegeben sind; in diesem Fall gilt⁶⁸: $\text{soistZielKostenA}_j(j, \text{siz}, k=5, \text{przl}, \text{ir}) = \text{wert}_{j_przl_prz}(j, \text{przl}, \text{Prz}_7) - i\text{Kosten}_j(j, \text{ir}, k=3)$.	305
soistZielOutputA_j(j,siz,k=1, przl,ir)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Ist-Zielabweichungsvektors (siz) zu dessen Komponente k=1 (d.h. der ersten Komponente) unter Berücksichtigung der Prozesszielliste (przl) des zugrunde liegenden Prozesses und dessen realisierten Reaktionsgrößenvektors (ir); der Funktionswert repräsentiert die Soll-Ist-Outputzielabweichung des hier relevanten Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 55 und 235 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Daten über den realisierten Output des zugrunde liegenden Prozesses gegeben sind; sodann gilt: $\text{soistZielOutputA}_j(j, \text{siz}, k=1, \text{przl}, \text{ir}) = 0$. Fall 2): liegt vor, wenn Istdaten über den realisierten Output des zugrunde liegenden Prozesses gegeben sind; in diesem Fall gilt⁶⁹: $\text{soistZielOutputA}_j(j, \text{siz}, k=1, \text{przl}, \text{ir}) = \text{wert}_{j_przl_prz}(j, \text{przl}, \text{Prz}_1) - i\text{Output}_j(j, \text{ir}, k=1)$.	301

⁶⁸ Wie bei SNr. 48 angemerkt, repräsentiert die Individuenkonstante „Prz₇“ das Prozesskostenziel.

⁶⁹ Wie bei SNr. 48 angemerkt, repräsentiert die Individuenkonstante „Prz₁“ das Prozessoutputziel.

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
soistZielQualA_j(j,siz,k=2,przl,rze)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Ist-Zielabweichungsvektors (siz) zu dessen Komponente k=2 (d.h. der zweiten Komponente) unter Berücksichtigung der Prozesszielliste (przl) des zugrunde liegenden Prozesses und dessen realisierten Prozesszielerreichungsvektors (rze); der Funktionswert repräsentiert die Soll-Ist-Qualitätszielabweichung des hier relevanten Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 55 und 291 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Daten über die realisierte Qualität des zugrunde liegenden Prozesses gegeben sind; sodann gilt: $\text{soistZielQualA}_j(j, \text{siz}, k=2, \text{przl}, \text{rze}) := 0$. Fall 2): andernfalls [d.h. Fall 1) liegt nicht vor] gilt⁷⁰: $\text{soistZielQualA}_j(j, \text{siz}, k=2, \text{przl}, \text{rze}) := \text{wert}_{j_przl_prz}(j, \text{przl}, \text{Prz}_2) - i\text{ZielEqual}_j(j, \text{rze}, k=2)$.	302
SoIstZielVektor(siz)	Etwas (siz) ist ein Soll-Ist-Zielabweichungsvektor.	295
soistZielWertsA_j(j,siz,k=6,przl,ir)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Ist-Zielabweichungsvektors (siz) zu dessen Komponente k=6 (der sechsten Komponente) unter Berücksichtigung der Prozesszielliste (przl) des zugrunde liegenden Prozesses und dessen realisierten Reaktionsgrößenvektors (ir); der Funktionswert repräsentiert die Soll-Ist-Wertschöpfungszielabweichung des hier relevanten Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionserte werden mithilfe der in SNr. 55 und 238 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Daten über die realisierte Wertschöpfung des zugrunde liegenden Prozesses gegeben sind; sodann gilt: $\text{soistZielWertsA}_j(j, \text{siz}, k=6, \text{przl}, \text{ir}) := 0$. Fall 2): liegt vor, wenn Daten über die realisierte Wertschöpfung des zugrunde liegenden Prozesses gegeben sind; in diesem Fall gilt⁷¹: $\text{soistZielWertsA}_j(j, \text{siz}, k=6, \text{przl}, \text{ir}) := \text{wert}_{j_przl_prz}(j, \text{przl}, \text{Prz}_8) - i\text{Werts}_j(j, \text{ir}, k=4)$.	306

⁷⁰ Wie bei SNr. 48 angemerkt, repräsentiert die Individuenkonstante „Prz₂“ das Prozessqualitätsziel.

⁷¹ Wie bei SNr. 48 angemerkt, repräsentiert die Individuenkonstante „Prz₈“ das Prozesswertschöpfungsziel.

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
SoIstZV_Komp(siz,k)	Ein Soll-Ist-Zielabweichungsvektor (siz) weist die Komponente (k) auf.	296
sOutput_j(j,sr,k=1)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines simulierten Reaktionsgrößenvektors (sr) zu dessen Komponente k=1 (d.h. der ersten Komponente); dieser Wert repräsentiert den gegebenenfalls vorliegenden simulierten Output des zugrunde liegenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten Prozesse gegebenenfalls vorliegenden Simulationsdaten festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Simulationsdaten über den simulierten Output (kurz <i>so</i>) für den hier relevanten Reaktionsgrößenvektor gegeben sind; sodann gilt: $sOutput_j(j,sr,k=1) := -1$. Fall 2): liegt vor, wenn Simulationsdaten für den hier relevanten Reaktionsgrößenvektor über den entsprechend simulierten Output <i>so</i> gegeben sind, sodann gilt: $sOutput_j(j,sr,k=1) := so$, mit $so \in \mathbb{N}_0$.	170

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
sowiawert_{j_pr_bppzl}(j,pr,bppzl,bp,pz)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt den Soll-Wird-Abweichungswert bezüglich der Beitragsposition (bp), die dem Projektziel (pz) zugeordnet ist und in der Liste Projektzielen zugeordneter Beitragspositionen (bppzl) enthalten ist, welche für die Prozessregel (pr) spezifiziert ist, die im Prozess (j) zur Anwendung vorgesehen ist. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 25, 280 und 281 spezifizierten Funktionsausdrücke festgelegt. Dabei werden drei Fälle unterschieden. Fall 1): liegt vor, falls die Individuenkonstante „None“ für das Projektziel (pz) gegeben ist oder der zugehörige Funktionswert von pzSkalaKNr(pz) größer als „1“ ist, sodann gilt: sowiawert_{j_pr_bppzl}(j,pr,bppzl,bp,pz):= nd („nd“ steht hier für „nicht definiert“). Fall 2): liegt vor, falls Fall 1) nicht zutrifft und für das jeweils relevante Projektziel (pz) der Funktionswert von pzSkalaKNr(pz) „1“ beträgt und noch keine Simulationsdaten oder noch keine Realisationsdaten über den Wert der zugrunde liegenden Beitragsposition gegeben sind; sodann gilt: sowiawert_{j_pr_bppzl}(j,pr,bppzl,bp,pz):= nv („nv“ steht hier für „no value“ bzw. „kein Wert“). Fall 3): liegt vor, falls Fall 1) nicht zutrifft und für das jeweils relevante Projektziel (pz) der Funktionswert von pzSkalaKNr(pz) „1“ beträgt und sowohl Simulationsdaten als auch Realisationsdaten über den Wert der zugrunde liegenden Beitragsposition gegeben sind; sodann gilt: sowiawert_{j_pr_bppzl}(j,pr,bppzl,bp,pz):= pbwert_{j_pr_bppzl}(j,pr,bppzl,bp,pz) – sbwert_{j_pr_bppzl}(j,pr,bppzl,bp,pz).	282
SoWiPRVektor(swpr)	Etwas (swpr) ist ein Soll-Wird-Potenzialreaktionsabweichungsvektor.	183
SoWiPRV_Komp(swpr,k)	Ein Soll-Wird-Potenzialreaktionsabweichungsvektor (swpr) weist die Komponente (k) auf.	184

Offener Funktionsausdruck	Bedeutung	SNr.
sowiPRVKomp_j(j, swpr,k,ppr,spr)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Wird-Potenzialreaktionsabweichungsvektors (swpr) zu dessen Komponente (k) unter Berücksichtigung des Wertes der geplanten Potenzialreaktion, die im Planpotenzialreaktionsvektor (ppr) des hier zugrunde liegenden Prozesses in der übereinstimmenden Komponente (k) als Sollgröße angegeben ist, sowie des Wertes der simulierten Potenzialreaktion, die im simulierten Potenzialreaktionsvektor (spr) des hier relevanten Prozesses in der wiederum übereinstimmenden Komponente (k) als Wirdgröße angegeben ist⁷². <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 90 und 182 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn zu der hier relevanten Wirdgröße noch keine Simulationsdaten gegeben sind; sodann gilt: $\text{sowiPRVKomp}_j(j, \text{swpr}, k, \text{ppr}, \text{spr}) := 0.$ Fall 2): liegt vor, wenn zu der hier relevanten Potenzialreaktion Simulationsdaten gegeben sind; sodann gilt: $\text{sowiPRVKomp}_j(j, \text{swpr}, k, \text{ppr}, \text{spr}) :=$ $\text{pPReaktVKomp}_j(j, \text{ppr}, k) -$ $\text{sPReaktVKomp}_j(j, \text{spr}, k).$	189
sowiATA_j(j,swa, k=1)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Wird-Abweichungsgrößenvektors (swa) zu dessen Komponente k=1 (d.h. der ersten Komponente); dieser Wert repräsentiert die Soll-Wird-Anfangsterminabweichung des zugrunde liegenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 99 und 108 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $\text{pATermin}_j(j) = -1$ (d.h., es liegt noch kein Plananfangstermin für den hier relevanten Prozess vor) oder wenn gilt: $\text{sATermin}_j(j) = -1$ (d.h., es liegt noch kein simulierter Anfangstermin für den hier relevanten Prozess vor), dann folgt daraus: $\text{sowiATA}_j(j, \text{swa}, k=1) := 0.$ Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $\text{sowiATA}_j(j, \text{swa}, k=1) := \text{pATermin}_j(j) - \text{sATermin}_j(j).$	174

⁷² Dabei wird unterstellt, dass in der Komponente (k) die Soll-Wird-Potenzialreaktionsabweichung der k-ten Potenzialart ($k=1, \dots, m$; $m \in \mathbb{N}$) verwaltet wird.

Offener Funktionsausdruck	Bedeutung	SNr.
sowiETA_j(j,swa,k=2)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Wird-Abweichungsgrößenvektors (swa) zu dessen Komponente k=2; dieser Wert repräsentiert die Soll-Wird-Endterminabweichung des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 107 und 123 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $pETermin_j(j) = -1$ (d.h., es liegt noch kein Planendtermin für den hier relevanten Prozess vor) oder wenn gilt: $sETermin_j(j) = -1$ (d.h., es liegt noch kein simulierter Endtermin für den hier relevanten Prozess vor), dann folgt daraus: $sowiETA_j(j,swa,k=2) := 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $sowiETA_j(j,swa,k=2) := pETermin_j(j) - sETermin_j(j)$.	175
sowiKA_j(j,swa,k=4,sr)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Wird-Abweichungsgrößenvektors (swa) zu dessen Komponente k=4 unter Berücksichtigung des zugehörigen simulierten prozessspezifischen Reaktionsgrößenvektors (sr); dieser Wert repräsentiert die Soll-Wird-Kostenabweichung des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 61 und 172 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $sKosten_j(j,sr,k=3) = -1$ (d.h., es liegt noch keine Simulationsdaten über die simulierten Kosten für den hier relevanten Prozess vor), dann folgt daraus: $sowiKA_j(j,swa,k=4,sr) := 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor; folgt: $sowiKA_j(j,swa,k=4,sr) := pKosten_j(j) - sKosten_j(j,sr,k=3)$.	177

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
sowiOpA_j(j, swa,k=3,sr)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Wird-Abweichungsgrößenvektors (swa) zu dessen Komponente k=3 unter Berücksichtigung des zugehörigen simulierten prozessspezifischen Reaktionsgrößenvektors (sr); dieser Wert repräsentiert die Soll-Wird-Outputabweichung des zugrunde liegenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 60 und 170 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $sOutput_j(j,sr,k=1) = -1$ (d.h., es liegen noch keine Simulationsdaten über den simulierten Output für den hier relevanten Prozess vor), dann folgt daraus: $sowiOpA_j(j,swa,k=3,sr) = 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor; folgt: $sowiOpA_j(j,swa,k=3,sr) = pOutput_j(j) - sOutput_j(j,sr,k=1)$.	176
SoWiV(swa)	Etwas (swa) ist ein Soll-Wird-Abweichungsgrößenvektor.	167
SoWiV_Komp(swa,k)	Ein Soll-Wird-Abweichungsgrößenvektor (swa) weist die Komponente (k) auf.	169
sowiWsA_j(j,swa,k=5,sr)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Wird-Abweichungsgrößenvektors (swa) zu dessen Komponente k=:5 unter Berücksichtigung des zugehörigen simulierten prozessspezifischen Reaktionsgrößenvektors (sr); dieser Wert repräsentiert die Soll-Wird-Wertschöpfungsabweichung des zugrunde liegenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 62 und 173 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $sWerts_j(j,sr,k=4) = -1$ (d.h., es liegen noch keine Simulationsdaten über die simulierte Wertschöpfung für den hier relevanten Prozess vor), dann folgt daraus: $sowiWsA_j(j,swa,k=5,sr) = 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $sowiWsA_j(j,swa,k=5,sr) = pWerts_j(j) - sWerts_j(j,sr,k=4)$.	178

Offener Funktionsausdruck	Bedeutung	SNr.
sowiZielATA_j(j,swz,k=3,przl)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Wird-Zielabweichungsvektors (swz) zu dessen Komponente k=3 (der dritten Komponente) unter Berücksichtigung der Prozesszielliste (przl) des zugrunde liegenden Prozesses; der Funktionswert repräsentiert die Soll-Wird-Anfangsterminzielabweichung des relevanten Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 55 und 108 spezifizierten Funktionsausdrücke festgelegt. Dabei werden drei Fälle unterschieden⁷³. Fall 1): liegt vor, wenn noch keine Daten über den simulierten Anfangstermin des jeweiligen Prozesses gegeben sind oder Daten über den simulierten Anfangstermin vorhanden sind und folgende Bedingung erfüllt ist: $\text{wert}_{j_przl_prz}(j,przl,Prz_3) \leq sATermin_j(j) \leq \text{wert}_{j_przl_prz}(j,przl,Prz_4),$ dann liegt keine (signifikante) Terminabweichung vor, dann folgt: $\text{sowiZielATA}_j(j,swz,k=3,przl) := 0$ Fall 2): liegt vor, falls Daten über den simulierten Anfangstermin gegeben sind und folgende Bedingung erfüllt ist: $sATermin_j(j) < \text{wert}_{j_przl_prz}(j,przl,Prz_3),$ dann liegt eine Terminverfrühung vor und es folgt: $\text{sowiZielATA}_j(j,swz,k=3,przl) := \text{wert}_{j_przl_prz}(j,przl,Prz_3) - sATermin_j(j)$ Fall 3): liegt vor, falls Daten über den simulierten Anfangstermin gegeben sind und folgende Bedingung erfüllt ist: $sATermin_j(j) > \text{wert}_{j_przl_prz}(j,przl,Prz_4),$ dann liegt eine Terminverspätung vor und es folgt: $\text{sowiZielATA}_j(j,swz,k=3,przl) := \text{wert}_{j_przl_prz}(j,przl,Prz_4) - sATermin_j(j).$	201

⁷³ Wie bei SNr. 48 angemerkt, repräsentieren im Folgenden die Individuenkonstante „Prz₃“ das Ziel des frühesten Anfangstermins und die Individuenkonstante „Prz₄“ das Ziel des spätesten Anfangstermins (entsprechend den Prozesszielen *pcz₃* und *pcz₄* innerhalb der Spezifikation von Parameter *dp₆(up)*).

Offener Funktionsausdruck	Bedeutung	SNr.
sowiZielETA_j(j,swz,k=4,przl)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Wird-Zielabweichungsvektors (swz) zu dessen Komponente k=4 (der vierten Komponente) unter Berücksichtigung der Prozesszielliste (przl) des zugrunde liegenden Prozesses; der Funktionswert repräsentiert die Soll-Wird-Endterminzielabweichung des betreffenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 55 und 123 spezifizierten Funktionsausdrücke festgelegt. Dabei werden drei Fälle unterschieden⁷⁴. Fall 1): liegt vor, wenn noch keine Daten über den simulierten Endtermin des jeweiligen Prozesses gegeben sind oder Daten über den simulierten Endtermin vorhanden sind und folgende Bedingung erfüllt ist: $\text{wert}_{j_przl_prz}(j,przl,Prz_5) \leq \text{sETermin}_j(j) \leq \text{wert}_{j_przl_prz}(j,przl,Prz_6),$ dann liegt keine (signifikante) Terminabweichung vor, dann folgt: $\text{sowiZielETA}_j(j,swz,k=4,przl) := 0$ Fall 2): liegt vor, falls Daten über den simulierten Endtermin gegeben sind und folgende Bedingung erfüllt ist: $\text{sETermin}_j(j) < \text{wert}_{j_przl_prz}(j,przl,Prz_5),$ dann liegt eine Terminverfrühung vor und es folgt: $\text{sowiZielETA}_j(j,swz,k=4,przl) := \text{wert}_{j_przl_prz}(j,przl,Prz_5) - \text{sETermin}_j(j)$ Fall 3): liegt vor, falls Daten über den simulierten Endtermin gegeben sind und folgende Bedingung erfüllt ist: $\text{sETermin}_j(j) > \text{wert}_{j_przl_prz}(j,przl,Prz_6),$ dann liegt eine Terminverspätung vor und es folgt: $\text{sowiZielETA}_j(j,swz,k=4,przl) := \text{wert}_{j_przl_prz}(j,przl,Prz_6) - \text{sETermin}_j(j).$	202

⁷⁴ Wie bei SNr. 48 angemerkt, repräsentiert im Folgenden die Individuenkonstante „Prz₅“ das Ziel des frühesten Endtermins und die Individuenkonstante „Prz₆“ das Ziel des spätesten Endtermins (entsprechend den Prozesszielen pcz_5 und pcz_6 innerhalb der Spezifikation von Parameter $dp_6(up)$).

Offener Funktionsausdruck	Bedeutung	SNr.
sowiZielKostenA_j(j,swz, k=5,przl,sr)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Wird-Zielabweichungsvektors (swz) zu dessen Komponente k=5 (der fünften Komponente) unter Berücksichtigung der Prozesszielliste (przl) des zugrunde liegenden Prozesses und dessen simulierten Reaktionsgrößenvektors (sr); der Funktionswert repräsentiert die Soll-Wird-Kostenzielabweichung des hier relevanten Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 55 und 172 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Daten über die simulierten Kosten des zugrunde liegenden Prozesses gegeben sind; sodann gilt: $\text{sowiZielKostenA}_j(j, \text{swz}, k=5, \text{przl}, \text{sr}) := 0.$ Fall 2): liegt vor, wenn Daten über die simulierten Kosten des zugrunde liegenden Prozesses gegeben sind; in diesem Fall gilt⁷⁵: $\text{sowiZielKostenA}_j(j, \text{swz}, k=5, \text{przl}, \text{sr}) := \text{wert}_{j_przl_prz}(j, \text{przl}, \text{Prz}_7) - \text{sKosten}_j(j, \text{sr}, k=3).$	203
sowiZielOutputA_j(j,swz, k=1,przl,sr)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Wird-Zielabweichungsvektors (swz) zu dessen Komponente k=1 (d.h. der ersten Komponente) unter Berücksichtigung der Prozesszielliste (przl) des zugrunde liegenden Prozesses und dessen simulierten Reaktionsgrößenvektors (sr); der Funktionswert repräsentiert die Soll-Wird-Outputzielabweichung des hier relevanten Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 55 und 170 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Daten über den simulierten Output des zugrunde liegenden Prozesses gegeben sind; sodann gilt: $\text{sowiZielOutputA}_j(j, \text{swz}, k=1, \text{przl}, \text{sr}) := 0.$ Fall 2): liegt vor, wenn Simulationsdaten über den simulierten Output des zugrunde liegenden Prozesses gegeben sind; in diesem Fall gilt⁷⁶: $\text{sowiZielOutputA}_j(j, \text{swz}, k=1, \text{przl}, \text{sr}) := \text{wert}_{j_przl_prz}(j, \text{przl}, \text{Prz}_1) - \text{sOutput}_j(j, \text{sr}, k=1).$	199

⁷⁵ Wie bei SNr. 48 angemerkt, repräsentiert die Individuenkonstante „Prz₇“ das Prozesskostenziel.

⁷⁶ Wie bei SNr. 48 angemerkt, repräsentiert die Individuenkonstante „Prz₁“ das Prozessoutputziel.

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
sowiZielQualA_j(j,swz,k=2,przl,sze)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Wird-Zielabweichungsvektors (swz) zu dessen Komponente k=2 (d.h. der zweiten Komponente) unter Berücksichtigung der Prozesszielliste (przl) des zugrunde liegenden Prozesses und dessen simulierten Prozesszielerreichungsvektors (sze); der Funktionswert repräsentiert die Soll-Wild-Qualitätszielabweichung des hier relevanten Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 55 und 191 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Daten über den simulierte Qualität des zugrunde liegenden Prozesses gegeben sind; sodann gilt: $\text{sowiZielEqualA}_j(j, \text{swz}, k=2, \text{przl}, \text{sze}) := 0$. Fall 2): andernfalls [d.h. Fall 1) liegt nicht vor] gilt⁷⁷: $\text{sowiZielEqualA}_j(j, \text{swz}, k=2, \text{przl}, \text{sze}) := \text{wert}_{j_przl_prz}(j, \text{przl}, \text{Prz}_2) - \text{sZielEqual}_j(j, \text{sze}, k=2)$.	200
SoWiZielVektor(swz)	Etwas (swz) ist ein Soll-Wird-Zielabweichungsvektor.	196
sowiZielWertsA_j(j,swz,k=6,przl,sr)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Soll-Wird-Zielabweichungsvektors (swz) zu dessen Komponente k=6 (der sechsten Komponente) unter Berücksichtigung der Prozesszielliste (przl) des zugrunde liegenden Prozesses und dessen simulierten Reaktionsgrößenvektors (sr); der Funktionswert repräsentiert die Soll-Wird-Wertschöpfungszielabweichung des hier relevanten Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 55 und 173 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Daten über die simulierte Wertschöpfung des zugrunde liegenden Prozesses gegeben sind; sodann gilt: $\text{sowiZielWertsA}_j(j, \text{swz}, k=6, \text{przl}, \text{sr}) := 0$. Fall 2): liegt vor, wenn Daten über die simulierte Wertschöpfung des zugrunde liegenden Prozesses gegeben sind; in diesem Fall gilt⁷⁸: $\text{sowiZielWertsA}_j(j, \text{swz}, k=6, \text{przl}, \text{sr}) := \text{wert}_{j_przl_prz}(j, \text{przl}, \text{Prz}_8) - \text{sWerts}_j(j, \text{sr}, k=4)$.	204

⁷⁷ Wie bei SNr. 48 angemerkt, repräsentiert die Individuenkonstante „Prz₂“ das Prozessqualitätsziel.

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
SoWiZV_Komp(swz,k)	Ein Soll-Wird-Zielabweichungsvektor (swz) weist die Komponente (k) auf.	197
spATermin_j(j)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt den spätesten Anfangstermin des Prozesses (j). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den betreffenden Prozessen über Annahmen im relevanten (Teil-) Projektkontext jeweils in Form einer nicht negativen reellen Zahl zugeordnet, dabei muss die Bedingung erfüllt sein, dass dieser Termin zeitlich nicht vor dem frühesten Anfangstermin des zugrunde liegenden Prozesses liegen darf; es gilt demnach: $\text{spATermin}(j) \geq \text{fATermin}(j).$	92
SPBestandV(spb)	Etwas (spb) ist ein simulierter entwicklungsprojektspezifischer (Wird-) Potenzialbestandsvektor.	365
SPBV_Komp(spb,k)	Ein simulierter entwicklungsprojektspezifischer (Wird-) Potenzialbestandsvektor (spb) weist die Komponente (k) auf.	366
spETermin_j(j)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt den spätesten Endtermin des Prozesses (j). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden den entsprechenden Prozessen (j) über Annahmen im relevanten (Teil-) Projektkontext jeweils in Form einer nicht negativen reellen Zahl direkt zugeordnet, wobei folgende Bedingung gilt: $\text{spETermin}(j) \geq \text{fETermin}(j).$	95

⁷⁸ Wie bei SNr. 48 angemerkt, repräsentiert die Individuenkonstante „Prz₈“ das Prozesswertschöpfungsziel.

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
sPReaktVKomp_j(j,spr,k)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert der gegebenenfalls simulierten Potenzialreaktion, die im zugehörigen (prozessspezifischen) simulierten Potenzialreaktionsvektor (spr) in dessen Komponente (k) anzugeben ist⁷⁹. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden anhand der für die hier relevanten Prozesse gegebenenfalls vorliegenden Simulationsdaten festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn zur hier relevanten Potenzialreaktion noch kein simulierter Wert (kurz <i>ws</i>) gegeben ist; sodann gilt: $\text{sPReaktVKomp}_j(j, \text{spr}, k) := 0$ Fall 2): liegt vor, wenn zur hier relevanten Potenzialreaktion ein simulierter Wert (<i>ws</i>) gegeben ist; sodann gilt: $\text{sPReaktVKomp}_j(j, \text{spr}, k) := ws, ws \in \mathbb{R}.$	182
SProzZielEVektor(sze)	Etwas (sze) ist ein simulierter Prozesszielerreichungsvektor.	186
SProzZielEV_Komp(sze,k)	Ein simulierter Prozesszielerreichungsvektor (sze) weist die Komponente (k) auf.	187
SVPBestandV(svpb)	Etwas (svpb) ist ein simulierter entwicklungsprojektspezifischer (Wird-) Verbrauchspotenzialbestandsvektor.	348
SVPBV_Komp(svpb,k)	Ein simulierter entwicklungsprojektspezifischer (Wird-) Verbrauchspotenzialbestandsvektor (svpb) weist die Komponente (k) auf.	351

⁷⁹ Dabei wird unterstellt, dass in der Komponente (k) die simulierte Potenzialreaktion der *k*-ten Potenzialart ($k=1, \dots, m; m \in \mathbb{N}$) verwaltet wird.

Offener Funktionsausdruck	Bedeutung	SNr.
sWerts_j(j,sr,k=4)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines simulierten Reaktionsgrößenvektors (sr) zu dessen Komponente k=4; dieser Wert repräsentiert die gegebenenfalls vorliegende simulierte Wertschöpfung des zugrunde liegenden Prozesses (bzw. die simulierte Prozessleistung). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten Prozesse gegebenenfalls vorliegenden Simulationsdaten festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Simulationsdaten über die simulierte Wertschöpfung (kurz <i>sw</i>) für den hier relevanten Reaktionsgrößenvektor gegeben sind; sodann gilt: sWerts_j(j,sr,k=4):= -1. Fall 2): liegt vor, wenn Simulationsdaten für den hier relevanten Reaktionsgrößenvektor über die entsprechende simulierte Wertschöpfung <i>sw</i> gegeben sind, sodann gilt: sWerts_j(j,sr,k=4):= <i>sw</i>, mit $sw \in \mathbb{R}_+$.	173
sZielEAT_j(j,sze,k=3)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines simulierten Prozesszielerreichungsvektors (sze) zu dessen Komponente k=3 (d.h. der dritten Komponente); dieser Wert repräsentiert den gegebenenfalls vorliegenden simulierten Anfangstermin des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe des in SNr. 108 spezifizierten Funktionsausdrucks wie folgt festgelegt: sZielEAT_j(j,sze,k=3):= sATermin_j(j).	192
sZielEET_j(j,sze,k=4)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines simulierten Prozesszielerreichungsvektors (sze) zu dessen Komponente k=4 (d.h. der vierten Komponente); dieser Wert repräsentiert den gegebenenfalls vorliegenden simulierten Endtermin des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe des in SNr. 123 spezifizierten Funktionsausdrucks wie folgt festgelegt: sZielEET_j(j,sze,k=4):= sETermin_j(j).	193

Offener Funktionsausdruck	Bedeutung	SNr.
sZielEKosten_j(j,sze,k=5,sr)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines simulierten Prozesszielerreichungsvektors (sze) zu dessen Komponente k=5 (d.h. der fünften Komponente) unter Berücksichtigung des simulierten Reaktionsgrößenvektors (sr) des zugrunde liegenden Prozesses; der Funktionswert repräsentiert die gegebenenfalls vorliegenden simulierten Kosten des zugrunde liegenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe des in SNr. 172 spezifizierten Funktionsausdrucks wie folgt festgelegt: $sZielEKosten_j(j,sze,k=5,sr) := sKosten_j(j,sr,k=4)$.	194
sZielEOutput_j(j,sze,k=1,sr)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines simulierten Prozesszielerreichungsvektors (sze) zu dessen Komponente k=1 (d.h. der ersten Komponente) unter Berücksichtigung des simulierten Reaktionsgrößenvektors (sr) des zugrunde liegenden Prozesses; der Funktionswert repräsentiert den gegebenenfalls vorliegenden simulierten Output des zugrunde liegenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe des in SNr. 170 spezifizierten Funktionsausdrucks wie folgt festgelegt: $sZielEOutput_j(j,sze,k=1,sr) := sOutput_j(j,sr,k=1)$.	190
sZielEQual_j(j,sze,k=2)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines simulierten Prozesszielerreichungsvektors (sze) zu dessen Komponente k=2 (d.h. der zweiten Komponente); dieser Wert repräsentiert die gegebenenfalls vorliegende simulierte Qualität des zugrunde liegenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden direkt anhand der für die hier relevanten Prozesse gegebenenfalls vorliegenden Simulationsdaten festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Simulationsdaten über die simulierte Qualität (kurz <i>sq</i>) für den hier relevanten simulierten Prozesszielerreichungsvektor gegeben sind; sodann gilt: $sZielEQual_j(j,sze,k=2) := -1$. Fall 2): liegt vor, wenn Simulationsdaten für den hier relevanten simulierten Prozesszielerreichungsvektors über die entsprechend simulierte Qualität <i>sq</i> gegeben sind, sodann gilt: $sZielEQual_j(j,sze,k=2) := sq$, mit $sq \in [1, 6]$.	191

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
sZielEWerts_j(j,sze,k=6,sr)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines simulierten Prozesszielerreichungsvektors (sze) zu dessen Komponente k=6 (d.h. der sechsten Komponente) unter Berücksichtigung des simulierten Reaktionsgrößenvektors (sr) des zugrunde liegenden Prozesses; der Funktionswert repräsentiert die gegebenenfalls vorliegende simulierte Wertschöpfung des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe des in SNr. 173 spezifizierten Funktionsausdrucks wie folgt festgelegt: sZielEWerts_j(j,sze,k=6,sr):= sWerts_j(j,sr,k=5).	195
UZListe(uzl)	Etwas (uzl) ist eine Liste von (Entscheidungs-) Umfeldzuständen.	43
UZL_Uz(uzl,zu)	Eine Liste von (Entscheidungs-) Umfeldzuständen (uzl) beinhaltet einen Umfeldzustand (zu).	45
Uzustand(zu)	Etwas (zu) ist ein (Entscheidungs-) Umfeldzustand.	44
valiPErgebnis_{j_pr_bppzl}(j,pr,bppzl)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) das Validitätsprüfungsergebnis seiner (Prozess-) Regel (pr) unter Berücksichtigung ihrer Liste Projektzielen zugeordneter Beitragspositionen (bppzl). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in den in SNr. 386 bis 388 spezifizierten wff enthaltenen Inferenzregeln in Form des Modus ponens als Konsequenz in der zugehörigen Implikation zugewiesen. Dabei werden drei Fälle unterschieden. Fall 1): liegt vor, wenn die Antezedenz der Implikation der unter SNr. 386 spezifizierten wff erfüllt ist, in diesem Fall gilt: valiPErgebnis_{j_pr_bppzl}(j,pr,bppzl):= TRUE. Fall 2): liegt vor, wenn die Antezedenz der Implikation der unter SNr. 387 spezifizierten wff erfüllt ist, in diesem Fall gilt: valiPErgebnis_{j_pr_bppzl}(j,pr,bppzl):= TotalFALSE. Fall 3): liegt vor, wenn die Antezedenz der Implikation der unter SNr. 388 spezifizierten wff erfüllt ist, in diesem Fall gilt: valiPErgebnis_{j_pr_bppzl}(j,pr,bppzl):= PartialFALSE.	385

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
verivErgebnis_{j_pr_avl}(j,pr,avl)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) das Verifikationsversuchsergebnis seiner (Prozess-) Regel (pr) unter Berücksichtigung ihrer Regelanwendungsvoraussetzungsliste (avl). <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in den in SNr. 320 bis 323 spezifizierten wff enthaltenen Inferenzregeln in Form des Modus ponens als Konsequenz in der zugehörigen Implikation zugewiesen. Dabei werden vier Fälle unterschieden. Fall 1): liegt vor, wenn die Antezedenz der Implikation der unter SNr. 320 spezifizierten wff erfüllt ist, in diesem Fall gilt: $\text{verivErgebnis}_{j_pr_avl}(j,pr,avl) = \text{TotalTRUE}$. Fall 2): liegt vor, wenn die Antezedenz der Implikation der unter SNr. 321 spezifizierten wff erfüllt ist, in diesem Fall gilt: $\text{verivErgebnis}_{j_pr_avl}(j,pr,avl) = \text{TotalFALSE}$. Fall 3): liegt vor, wenn die Antezedenz der Implikation der unter SNr. 322 spezifizierten wff erfüllt ist, in diesem Fall gilt: $\text{verivErgebnis}_{j_pr_avl}(j,pr,avl) = \text{PartialTRUE}$. Fall 4): liegt vor, wenn die Antezedenz der Implikation der unter SNr. 323 spezifizierten wff erfüllt ist, in diesem Fall gilt: $\text{verivErgebnis}_{j_pr_avl}(j,pr,avl) = \text{PartialFALSE}$.	319
VorgängerProzNr(vpn)	Etwas (vpn) ist eine Vorgängerprozessnummer.	31
VorProzNrListe(vpnl)	Etwas (vpnl) ist eine Vorgängerprozessnummernliste.	29
vpBedarfVKomp_j(j,vp,k)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert des Bedarfs an Verbrauchspotenzial, das im zugehörigen (prozessspezifischen) Verbrauchspotenzialbedarfsvektor (vp) in dessen Komponente (k) anzugeben ist. <u>Funktionsgleichung</u>: Die Funktionswerte werden direkt über Annahmen für die relevanten Prozesse in Form nicht negativer reeller Zahlen zugeordnet.	86
VPBedarfVektor(vp)	Etwas (vp) ist ein Verbrauchspotenzialbedarfsvektor.	83
VPBedV_Komp(vp,k)	Ein Verbrauchspotenzialbedarfsvektor (vp) weist die Komponente (k) auf.	84

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
VPNListe_VProzNr(vpnl,vpn)	Etwas (vpn) beinhaltet eine Vorgängerprozessnummer (vpn).	32
wert_{j_przl_prz}(j,przl,prz)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert des zu seiner Prozesszielliste (przl) gehörenden Prozessziels (prz). <u>Funktionsgleichung:</u> Die Funktionswerte werden den einzelnen Prozesszielen (prz) der betreffenden Prozessliste über Annahmen für die zugrunde liegenden Prozesse (j) im jeweils relevanten (Teil-) Projektkontext in Form einer nicht negativen reellen Zahl zugeordnet; für Prozessziele, die (noch) nicht definiert bzw. spezifiziert sind⁸⁰, gilt: $\text{wert}_{\text{przl_prz}}(j,\text{przl},\text{prz}) := 0$.	55
wiistATA_j(j,wia,k=1)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Wird-Ist-Abweichungsgrößenvektors (wia) zu dessen Komponente k=1 (d.h. der ersten Komponente); dieser Wert repräsentiert die Wird-Ist-Anfangsterminabweichung des zugrunde liegenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 108 und 124 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $\text{sATermin}_j(j) = -1$ (d.h., es liegt noch kein simulierter Anfangstermin für den hier relevanten Prozess vor) oder wenn gilt: $\text{iATermin}_j(j) = -1$ (d.h., es liegt noch kein realisierter Anfangstermin für den hier relevanten Prozess vor), dann folgt daraus: $\text{wiistATA}_j(j,\text{wia},k=1) := 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $\text{wiistATA}_j(j,\text{wia},k=1) := \text{sATermin}_j(j) - \text{iATermin}_j(j)$.	274

⁸⁰ Dies entspricht der Vorgehensweise für (noch) nicht spezifizierte Prozessziele gem. Parameter $dp_6(up)$.

Offener Funktionsausdruck	Bedeutung	SNr.
wiistawert_{j_pr_bppzl}(j,pr,bppzl, bp,pz)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt den Wird-Ist-Abweichungswert bezüglich der Beitragsposition (bp), die dem Projektziel (pz) zugeordnet ist und in der Liste Projektzielen zugeordneter Beitragspositionen (bppzl) enthalten ist, welche für die Prozessregel (pr) spezifiziert ist, die im Prozess (j) zur Anwendung vorgesehen ist. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 280, 281 und 283 spezifizierten Funktionsausdrücke festgelegt. Dabei werden drei Fälle unterschieden. Fall 1): liegt vor, falls die Individuenkonstante „None“ für das Projektziel (pz) gegeben ist oder der Funktionswert von pzSkalaKNr(pz) größer als „1“ ist, sodann gilt: $wiistawert_{j_pr_bppzl}(j,pr,bppzl,bp,pz) := nd$ („nd“ steht hier für „nicht definiert“). Fall 2): liegt vor, falls Fall 1) nicht zutrifft und für das jeweils relevante Projektziel (pz) der Funktionswert von pzSkalaKNr(pz) „1“ beträgt und noch keine Simulationsdaten oder noch keine Realisationsdaten über den Wert der zugrunde liegenden Beitragsposition gegeben sind; sodann gilt: $wiistawert_{j_pr_bppzl}(j,pr,bppzl,bp,pz) := nv$ („nv“ steht hier für „no value“ bzw. „kein Wert“). Fall 3): liegt vor, falls Fall 1) nicht zutrifft und für das jeweils relevante Projektziel (pz) der Funktionswert von pzSkalaKNr(pz) „1“ beträgt und sowohl Simulationsdaten als auch Realisationsdaten über den Wert der zugrunde liegenden Beitragsposition gegeben sind; sodann gilt: $wiistawert_{j_pr_bppzl}(j,pr,bppzl,bp,pz) :=$ $sbwert_{j_pr_bppzl}(j,pr,bppzl,bp,pz) -$ $ibwert_{j_pr_bppzl}(j,pr,bppzl,bp,pz).$	285

Offener Funktionsausdruck	Bedeutung	SNr.
wiistETA_j(j,wia,k=2)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Wird-Ist-Abweichungsgrößenvektors (wia) zu dessen Komponente k=2; dieser Wert repräsentiert die Wird-Ist-Endterminabweichung des zugrunde liegenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 123 und 139 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $sETermin_j(j) = -1$ (d.h., es liegt noch kein simulierter Endtermin für den hier relevanten Prozess vor) oder wenn gilt: $iETermin_j(j) = -1$ (d.h., es liegt noch kein realisierter Endtermin für den hier relevanten Prozess vor), dann folgt daraus: $wiistETA_j(j,wia,k=2) := 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $wiistETA_j(j,wia,k=2) := sETermin_j(j) - iETermin_j(j)$.	275
wiistOpA_j(j,wia,k=3,sr,ir)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Wird-Ist-Abweichungsgrößenvektors (wia) zu dessen Komponente k=3 unter Berücksichtigung des zugehörigen simulierten prozessspezifischen Reaktionsgrößenvektors (sr) sowie des zugehörigen realisierten prozessspezifischen Reaktionsgrößenvektors (ir); dieser Wert repräsentiert die Wird-Ist-Outputabweichung des zugrunde liegenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 170 und 235 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $sOutput_j(j,sr,k=1) = -1$ (d.h., es liegen noch keine Daten über den simulierten Output für den hier relevanten Prozess vor) oder wenn gilt: $iOutput_j(j,ir,k=1) = -1$ (d.h., es liegen noch keine Istdaten über den realisierten Output für den hier relevanten Prozess vor), dann folgt daraus: $wiistOpA_j(j,wia,k=3,sr,ir) := 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $wiistOpA_j(j,wia,k=3,sr,ir) := sOutput_j(j,sr,k=1) - iOutput_j(j,ir,k=1)$.	276

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
wiistKA_j(j,wia,k=4,sr,ir)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines Wird-Ist-Abweichungsgrößenvektors (wia) zu dessen Komponente k=4 unter Berücksichtigung des zugehörigen simulierten prozessspezifischen Reaktionsgrößenvektors (sr) sowie des zugehörigen realisierten prozessspezifischen Reaktionsgrößenvektors (ir); dieser Wert repräsentiert die Wird-Ist-Kostenabweichung des zugrunde liegenden Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 172 und 237 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $sKosten_j(j,sr,k=3) = -1$ (d.h., es liegen noch keine Daten über die simulierten Kosten für den hier relevanten Prozess vor), oder wenn gilt: $iKosten_j(j,ir,k=3) = -1$ (d.h., es liegen noch keine Istdaten über die realisierten Kosten für den hier relevanten Prozess vor), dann folgt daraus: $wiistKA_j(j,wia,k=4,sr,ir) = 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $wiistKA_j(j,sia,k=4,sri,ir) = sKosten_j(j,sr,k=3) - iKosten_j(j,ir,k=3)$.	277
WiIstPRVektor(wipr)	Etwas (wipr) ist ein Wird-Ist-Potenzialreaktionsabweichungsvektor.	271
WiIstPRV_Komp(wipr,k)	Ein Wird-Ist-Potenzialreaktionsabweichungsvektor (wipr) weist die Komponente (k) auf.	272

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
wiistPRVKomp_j(j, wipr,k,spr,ipr)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines Wird-Ist-Potenzialreaktionsabweichungsvektors (wipr) zu dessen Komponente (k) unter Berücksichtigung des Wertes der simulierten Potenzialreaktion, die im simulierten Potenzialreaktionsvektor (spr) des hier zugrunde liegenden Prozesses in der übereinstimmenden Komponente (k) als Wirdgröße angegeben ist, sowie des Wertes der realisierten Potenzialreaktion, die im realisierten Potenzialreaktionsvektor (ipr) des hier relevanten Prozesses in der wiederum übereinstimmenden Komponente (k) als Istgröße angegeben ist⁸¹. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 182 und 247 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn zu der hier relevanten Wirdgröße noch keine Simulationsdaten oder zu der entsprechend relevanten Istgröße noch keine Realisationsdaten gegeben sind; sodann gilt: wiistPRVKomp_j(j,wipr,k,spr,ipr):= 0. Fall 2): liegt vor, wenn zu der hier relevanten Potenzialreaktion Simulationsdaten gegeben sind; sodann gilt: wiistPRVKomp_j(j,wipr,k,spr,ipr):= sPReaktVKomp_j(j,spr,k) – iPReaktVKomp_j(j,ipr,k).	279
WiIstV(wia)	Etwas (wia) ist ein Wird-Ist-Abweichungsgrößenvektor.	268
WiIstV_Komp(wia,k)	Ein Wird-Ist-Abweichungsgrößenvektor (wia) weist die Komponente (k) auf.	270

⁸¹ Dabei wird unterstellt, dass in der Komponente (k) die Wird-Ist-Potenzialreaktionsabweichung der k -ten Potenzialart ($k=1, \dots, m$; $m \in \mathbb{N}_{>0}$) verwaltet wird.

Offener Funktionsausdruck	Bedeutung	SNr.
wiistWsA_j(j,wia,k=5,sr,ir)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Wird-Ist-Abweichungsgrößenvektors (wia) zu dessen Komponente k=5 unter Berücksichtigung des zugehörigen simulierten prozessspezifischen Reaktionsgrößenvektors (sr) sowie des zugehörigen realisierten prozessspezifischen Reaktionsgrößenvektors (ir); dieser Wert repräsentiert die Wird-Ist-Wertschöpfungsabweichung des zugrunde liegenden Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 173 und 238 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden: Fall 1): liegt vor, wenn gilt: $sWerts_j(j,sr,k=4) = -1$ (d.h., es liegen noch keine Daten über die simulierte Wertschöpfung für den hier relevanten Prozess vor), oder wenn gilt: $iWerts_j(j,ir,k=4) = -1$ (d.h., es liegen noch keine Istdaten über die realisierte Wertschöpfung für den hier relevanten Prozess vor), dann folgt daraus: $wiistWsA_j(j,wia,k=5,si,ir) = 0$. Fall 2): andernfalls, d.h. Fall 1) liegt nicht vor, folgt: $wiistWsA_j(j,wia,k=5,sr,ir) = sWerts_j(j,sr,k=4) - iWerts_j(j,ir,k=4)$.	278
wiistZielATA_j(j,wiz,k=3)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Wird-Ist-Zielabweichungsvektors (wiz) zu dessen Komponente k=3 (der dritten Komponente); der Funktionswert repräsentiert die Wird-Ist-Anfangsterminzielabweichung des relevanten Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 108 und 124 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Daten über den simulierten Anfangstermin oder den realisierten Anfangstermin des jeweiligen Prozesses gegeben sind; dann gilt: $wiistZielATA_j(j,wiz,k=3) = 0$. Fall 2): andernfalls [d.h. Fall 1) liegt nicht vor] gilt: $wiistZielATA_j(j,wiz,k=3) = sATermin_j(j) - iATermin_j(j)$.	309

Offener Funktionsausdruck	Bedeutung	SNr.
wiistZielETA_j(j,wiz,k=4)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Wird-Ist-Zielabweichungsvektors (wiz) zu dessen Komponente k=4 (der vierten Komponente); der Funktionswert repräsentiert die Wird-Ist-Endterminzielabweichung des relevanten Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 123 und 139 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Daten über den simulierten Endtermin oder den realisierten Endtermin des jeweiligen Prozesses gegeben sind; dann gilt: $wiistZielETA_j(j,wiz,k=4) := 0$. Fall 2): andernfalls [d.h. Fall 1) liegt nicht vor] gilt: $wiistZielETA_j(j,wiz,k=4) := sETermin_j(j) - iETermin_j(j)$.	310
wiistZielKostenA_j(j,wiz,k=5, sr,ir)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Wird-Ist-Zielabweichungsvektors (wiz) zu dessen Komponente k=5 (der fünften Komponente) unter Berücksichtigung des simulierten Reaktionsgrößenvektors (sr) des zugrunde liegenden Prozesses und dessen realisierten Reaktionsgrößenvektors (ir); der Funktionswert repräsentiert die Wird-Ist-Kostenzielabweichung des hier relevanten Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 172 und 237 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Daten über die simulierten Kosten oder die realisierten Kosten des zugrunde liegenden Prozesses gegeben sind; sodann gilt: $wiistZielKostenA_j(j,wiz,k=5,sr,ir) := 0$. Fall 2): liegt vor, wenn Daten über die simulierten Kosten und die realisierten Kosten des zugrunde liegenden Prozesses vorliegen; in diesem Fall gilt: $wiistZielKostenA_j(j,wiz,k=5,sr,ir) := sKosten_j(j,sr,k=3) - iKosten_j(j,ir,k=3)$.	311

Offener Funktionsausdruck	Bedeutung	SNr.
wiistZielOutputA_j(j,wiz,k=1, sr,ir)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Wird-Ist-Zielabweichungsvektors (wiz) zu dessen Komponente k=1 (d.h. der ersten Komponente) unter Berücksichtigung des simulierten Reaktionsgrößenvektors (sr) des zugrunde liegenden Prozesses und dessen realisierten Reaktionsgrößenvektors (ir); der Funktionswert repräsentiert die Wird-Ist-Outputzielabweichung des hier relevanten Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 170 und 235 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Daten über den simulierten Output oder den realisierten Output des zugrunde liegenden Prozesses gegeben sind; sodann gilt: $\text{wiistZielOutputA}_j(j, \text{wiz}, k=1, \text{sr}, \text{ir}) := 0.$ Fall 2): liegt vor, wenn Daten über den simulierten und den realisierten Output des zugrunde liegenden Prozesses gegeben sind; in diesem Fall gilt: $\text{wiistZielOutputA}_j(j, \text{wiz}, k=1, \text{sr}, \text{ir}) := \text{sOutput}_j(j, \text{sr}, k=1) - \text{iOutput}_j(j, \text{ir}, k=1).$	307
wiistZielQualA_j(j,wiz,k=2, sze,rze)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den Prozess (j) den Wert seines Wird-Ist-Zielabweichungsvektors (wiz) zu dessen Komponente k=2 (d.h. der zweiten Komponente) unter Berücksichtigung der simulierten Prozesszielerreichungsvektors (sze) des zugrunde liegenden Prozesses und dessen realisierten Prozesszielerreichungsvektors (rze); der Funktionswert repräsentiert die Wird-Ist-Qualitätszielabweichung des hier relevanten Prozesses. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden mithilfe der in SNr. 191 und 291 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Daten über die simulierte oder die realisierte Qualität des zugrunde liegenden Prozesses gegeben sind; sodann gilt: $\text{wiistZielEQualA}_j(j, \text{wiz}, k=2, \text{sze}, \text{rze}) := 0.$ Fall 2): andernfalls [d.h. Fall 1) liegt nicht vor] gilt: $\text{wiistZielEQualA}_j(j, \text{wiz}, k=2, \text{sze}, \text{rze}) := \text{sZielEQual}_j(j, \text{sze}, k=2) - \text{iZielEQual}_j(j, \text{rze}, k=2).$	308

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
WilstZielVektor(wiz)	Etwas (swz) ist ein Wird-Ist-Zielabweichungsvektor.	298
wiistZielWertsA_j(j,wiz,k=6, sr,ir)	<u>Funktionsbeschreibung</u>: Diese Funktion ermittelt für den Prozess (j) den Wert seines Wird-Ist-Zielabweichungsvektors (wiz) zu dessen Komponente k=6 (der sechsten Komponente) unter Berücksichtigung des simulierten Reaktionsgrößenvektors (sr) des zugrunde liegenden Prozesses und dessen realisierten Reaktionsgrößenvektors (ir); der Funktionswert repräsentiert die Wird-Ist-Wertschöpfungszielabweichung des hier relevanten Prozesses. <u>Funktionsgleichung</u>: Die einzelnen Funktionswerte werden mithilfe der in SNr. 173 und 238 spezifizierten Funktionsausdrücke festgelegt. Dabei werden zwei Fälle unterschieden. Fall 1): liegt vor, wenn noch keine Daten über die simulierte Wertschöpfung oder die realisierte Wertschöpfung des zugrunde liegenden Prozesses gegeben sind; sodann gilt: $wiistZielWertsA_j(j,wiz,k=6,sr,ir) := 0$. Fall 2): liegt vor, wenn Daten über die simulierte Wertschöpfung und die realisierte Wertschöpfung des zugrunde liegenden Prozesses gegeben sind; in diesem Fall gilt: $wiistZielWertsA_j(j,wiz,k=6,sr,ir) := sWerts_j(j,sr,k=4) - iWerts_j(j,ir,k=4)$.	312
WilstZV_Komp(wiz,k)	Ein Wird-Ist-Zielabweichungsvektor (wiz) weist die Komponente (k) auf.	299

Offenes Atom bzw. offener Funktionsausdruck	Bedeutung	SNr.
zbart_{j_pr_bppzl}(j,pr,bppzl,bp,pz)	<u>Funktionsbeschreibung:</u> Diese Funktion ermittelt für den geplanten Wert der Beitragsposition (bp), die dem Projektziel (pz) zugeordnet ist und in der Liste Projektzielen zugeordneter Beitragspositionen (bppzl) enthalten ist, welche für die Prozessregel (pr) spezifiziert ist, die im Prozess (j) zur Anwendung vorgesehen ist, die Art des zugrunde liegenden Zielwertes. <u>Funktionsgleichung:</u> Die einzelnen Funktionswerte werden über Annahmen hinsichtlich der jeweiligen Beitragsposition (bp) und des zugehörigen Projektziels (pz) im zugrunde liegenden Projektkontext zugeordnet. Dabei werden drei Fälle unterschieden. Fall 1): liegt vor, wenn die Art des Zielwertes ein Fixwert ist, sodann gilt: zbart_{j_pr_bppzl}(j,pr,bppzl,bp,pz):= 0; Fall 2) liegt vor, wenn die Art des Zielwertes ein Mindestwert ist, sodann gilt: zbart_{j_pr_bppzl}(j,pr,bppzl,bp,pz):= 1; Fall 3) liegt vor, wenn die Art des Zielwertes ein Höchstwert ist, sodann gilt: zbart_{j_pr_bppzl}(j,pr,bppzl,bp,pz):= 2.	384
Zusatzkennzeichen(azk)	Etwas (azk) ist ein aktuelles Zusatzkennzeichen.	5

A-2 Anhang zur Simulationssprache Visual SLAM und zum Simulationssystem AweSim

A-2.1 Zulässige Operatoren auf der Ebene der graphischen Programmierung von Visual SLAM

Zulässige Operatoren	Definition
!	unäres NICHT
~	unäres Bit-Komplement
-	unäres Minus
*	Multiplikation
/	Division
%	Modulo
+	Addition
-	Subtraktion
<<	Bit-Verschiebung nach links
>>	Bit-Verschiebung nach rechts
<	kleiner als
<=	kleiner oder gleich
>	größer als
>=	größer oder gleich
==	gleich
!=	ungleich
&	UND-Verknüpfung von Bits
^	Exklusive-ODER-Verknüpfung von Bits
	ODER-Verknüpfung von Bits
&&	Logische UND-Verknüpfung
	Logische ODER-Verknüpfung

Tab. A-2.1: Zulässige Operatoren auf der Ebene der graphischen Programmierung von Visual SLAM⁸²

⁸² Entnommen aus Pütz 2004 /Produktionssysteme/ S. 591; in Anlehnung an Pritsker Corporation 1997 /Reference/ S. 5.

A-2.2 Netzwerkbezogene Variablen in Visual SLAM

(1) Einheitenspezifische Variablen

Variablenname	Definition
ATRIB[I]	reelles Attribut I der aktuellen Einheit
LTRIB[I]	ganzzahliges Attribut I der aktuellen Einheit
STRIB[I]	Zeichenketten-Attribut I der aktuellen Einheit
ENUM	eindeutiger ganzzahliger nur-Lese-Bezeichner der aktuellen Einheit
ERETURN	reelle einheitenspezifische Variable, deren Wert mit dem Wert des Parameters ERETURN des RETURNVSN-Knotens ⁸³ spezifiziert wird, an dem während eines Simulationslaufes aktuell eine Einheit eintrifft
ETYPE	ganzzahlige einheitenspezifische Variable zur Vergabe einer Typennummer für die aktuelle Einheit

Tab. A-2.2.1: Netzwerkbezogene einheitenspezifische Variablen in Visual SLAM⁸⁴

(2) Benutzerspezifizierbare globale Variablen

Variablenname	Definition
XX[I]	reelle globale Variable I
LL[I]	ganzzahlige globale Variable I
SZ[I]	globale Zeichenketten-Variable I
ARRAY[I,J]	Feldvariable J des Feldes I
SS[I]	globale Statusvariable I (zur kontinuierlichen Simulation)
DD[I]	1. Ableitung der Statusvariablen SS[I] mit $DD[I] = \frac{dSS[I]}{dt}$
MSTOP	mit dieser Variablen kann die aktuelle Simulation gestoppt werden, sofern ihr Wert auf „-1“ gesetzt wird
II	ganzzahlige globale Variable, II wird häufig als Index oder Argument eines Ausdrucks verwendet

Tab. A-2.2.2: Netzwerkbezogene benutzerspezifizierbare globale Variablenarten in Visual SLAM⁸⁵

⁸³ Hinsichtlich des (in objektorientierten Subnetzwerken) in Visual SLAM modellierbaren RETURNVSN-Knotens sei verwiesen auf Pritsker, O'Reilly 1999 /Visual SLAM/ S. 286; Pritsker Corporation 1997 /Reference/ S. 46; ferner Pütz 2004 /Produktionssysteme/ S 383 f.

⁸⁴ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 339; in Anlehnung Pritsker Corporation 1997 /Reference/ S. 6.

⁸⁵ Weitestgehend entnommen aus Pütz 2004 /Produktionssysteme/ S. 340; in Anlehnung Pritsker Corporation 1997 /Reference/ S. 6.

(3) Globale „nur-Lese“ Variablen in Visual SLAM

Variablenname	Definition
DTMIN	minimale kontinuierliche Schrittgröße gem. CONTINUOUS-Befehl
DTMAX	maximale kontinuierliche Schrittgröße gem. CONTINUOUS-Befehl
DTNOW	aktuelle kontinuierliche Schrittgröße gem. CONTINUOUS-Befehl
MATRIB	größter zulässiger ATRIB Index gem. Definition im LIMITS-Befehl
MLTRIB	größter zulässiger LTRIB Index gem. Definition im LIMITS-Befehl
MSTRIB	größter zulässiger STRIB Index gem. Definition im LIMITS-Befehl
MXXX	größter zulässiger LL Index gem. Definition im LIMITS-Befehl
MXLL	größter zulässiger LL Index gem. Definition im LIMITS-Befehl
MXSZ	größter zulässiger SZ Index gem. Definition im LIMITS-Befehl
NNCLT	größte (im aktuellen Szenario) verwendete COLCT Nummer
NNEQD	größter zulässiger DD Index gem. Definition im CONTINUOUS-Befehl
NNEQS	größter zulässiger SS Index gem. Definition im CONTINUOUS-Befehl
NNRNS	Anzahl der Simulationsläufe gem. Definition im GEN-Befehl
NNRUN	Nummer des aktuellen Simulationslaufs
NNSTA	größte verwendete TIMST Nummer (im aktuellen Szenario)
SCENARIO	Zeichenkette, die den Namen des aktuellen Szenarios beinhaltet
TNOW	aktuelle Simulationszeit
TTBEG	Startzeitpunkt der Simulation gem. Definition im INITIALIZE-Befehl
TTFIN	Startzeitpunkt der Simulation gem. Definition im INITIALIZE-Befehl

Tab. A-2.2.3: Liste sämtlicher globaler 'nur-Lese' Variablen in Visual SLAM⁸⁶**A-2.3 Grundlegende Netzwerkelemente in Visual SLAM**

Zu den im Folgenden angeführten graphischen Netzwerksymbolen wird, soweit in Visual SLAM definiert, jeweils eine kommaseparierte Parameterliste der zugehörigen Visual SLAM Programmanweisung mit entsprechenden Erläuterungen angeführt⁸⁷. Für ein adäquates Verständnis dieser Angaben ist es erforderlich, die in Visual SLAM für Parameterspezifikationen zulässigen Ausdrücke, welche aus Klammern und Visual SLAM Sprachelementen bestehen, beispielhaft darzustellen; dazu dient die nachfolgende Tabelle.

⁸⁶ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 592; in Anlehnung an Pritsker Corporation 1997 /Reference/ S. 7, Table 2.4.

⁸⁷ Die im Folgenden angeführten Netzwerksymbole nebst Erläuterungen sind weitestgehend entnommen aus Pütz 2004 /Produktionssysteme/ S. 342 ff. u. S. 593 ff.

Ausdrucksart	Erläuterung
Konstanter Ausdruck	Numerischer Ausdruck, der keinerlei globale Funktionen oder Variablen enthält. Beispiel: $1+(4*12)$
Globaler Ausdruck	Numerischer Ausdruck, der keine einheitenspezifische Variablen enthält. Beispiel: $20.3*XX[(2+LL[3])]$
Globaler linksseitiger Ausdruck	Benutzerspezifisierbare (globale) Variable, deren Index aus einem globalen Ausdruck besteht. Beispiel: $XX[(1+LL[3])]$
Linksseitiger Ausdruck	Benutzerspezifisierbare (globale) oder einheitenspezifische Variable, deren Index aus einem (numerischen) beliebigen Ausdruck besteht. Ein linksseitiger Ausdruck, der entweder auf benutzerspezifisierbare (globale) oder auf einheitenspezifische Variablen beschränkt ist, möge im Folgenden als benutzerspezifisierbarer oder einheitenspezifischer linksseitiger Ausdruck bezeichnet werden. Beispiel: $ATTRIB[1+LL[3]]+LTRIB[4]$
Beliebiger Ausdruck	Jede zulässige Kombination aus Visual SLAM Sprachelementen mit numerischem Wert. Beispiel: $14.3*XX[(2+LL[3])] * ABS(ATTRIB[3])$

Tab. A-2.3.1: Zulässige Ausdrucksarten in Visual SLAM⁸⁸

A-2.3.1 Blöcke

(1) RESOURCE-Block

<u>Netzwerksymbol:</u> RNUM RLBL CAP IFL repeats		<u>Programmanweisung in Visual SLAM:</u> RESOURCE,RNUM,RLBL,CAP,{IFL,repeats};
Eingabefeld	Option	Voreinstellung
RNUM	positiver ganzzahliger konstanter Ausdruck	nächster sequentieller Index
RLBL	eindeutig identifizierbare Zeichenkette	Angabe erforderlich
CAP	nicht-negativer ganzzahliger globaler Ausdruck	1
IFL, repeats	positiver ganzzahliger konstanter Ausdruck	Angabe erforderlich

Abb. A-2.3.1.1: RESOURCE-Block mit Parameterliste und Erläuterungen⁸⁹

⁸⁸ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 346; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 3 f. Wie im oben angeführten Beispiel eines globalen Ausdrucks ersichtlich, sei explizit darauf hingewiesen, dass in Visual SLAM als Dezimaltrennzeichen anstelle des Kommas der Punkt zu verwenden ist.

Der RESSOURCE-Block dient zur Definition einer Ressource bzw. eines Engpasses „RLBL“ mit der Anfangskapazität „CAP“. Jedem RESSOURCE-Block wird von Visual SLAM ein numerischer Schlüssel (Nummer) zugeordnet, der benutzerspezifisch mit dem Parameter „RNUM“ modifiziert werden kann. In einem RESSOURCE Block ist mindestens eine Datei unter Angabe einer zugehörigen Dateinummer „IFL“ festzulegen, die als Warteschlange für Einheiten dient, solange deren Kapazitätsanforderungen an „RLBL“ nicht erfüllt werden können. Zu diesem Zweck kann auch eine Liste von m Warteschlangen-Dateien einer Ressource zugeordnet werden, wobei über die Listenposition eine Reihenfolge der Abarbeitung⁹⁰ der Kapazitätsanforderungen implementiert werden kann⁹¹.

Die Kapazitätsanforderungen an die mit Hilfe von RESOURCE-Blöcken definierten Ressourcen werden unter Angabe der entsprechenden Warteschlangendateien mit Hilfe von AWAIT- oder PREEMPT-Knoten implementiert⁹². Dabei werden für Einheiten, die an diesem Knoten eintreffen, die im jeweiligen Knoten angeforderten und zugleich verfügbaren Kapazitätseinheiten belegt, die mit FREE-Knoten⁹³ wieder freigegeben werden können.

⁸⁹ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 346; vgl. hierzu Pritsker Corporation 1997 /Reference/ S. 44 f.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 164 ff.

⁹⁰ Dabei erfolgt eine Priorisierung der Dateien listenpositionsspezifisch von oben nach unten; im BLOCK-Symbol entsprechend ihrer Positionierung von links nach rechts. Vgl. hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 165 f. u. S. 204.

⁹¹ Vgl. hierzu auch die Erläuterungen zum RESSOURCE-Block in SLAM II in Witte, Claus, Helling 1994 /SLAM/ S. 247.

⁹² Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 166 ff. u. S. 194 ff. Vgl. ferner die Erläuterungen zum AWAIT- und PREEMPT-Knoten in (3) und (4) in Abschnitt A-2.3.3.

⁹³ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 170 ff. Vgl. ferner die Erläuterungen zum FREE-Knoten in (5) in Abschnitt A-2.3.3.

(2) GROUP-Block

<u>Netzwerksymbol:</u>				<u>Programmanweisung in Visual SLAM:</u>
GRNUM	GRLBL	{RESNUM, repeats}	GRSELRULE	GROUP,GRNUM,{RESNUM, repeats},GRSELRULE;
Eingabefeld	Option			Voreinstellung
GRNUM	positiver ganzzahliger konstanter Ausdruck			nächster sequentieller Index
GRLBL	eindeutig identifizierbare Zeichenkette			Angabe erforderlich
RESNUM, repeats	positiver ganzzahliger konstanter Ausdruck oder Zeichenkette, die auf einen RESOURCE-Block referenziert			Angabe erforderlich
GRSELRULE	ORDER, CYCLIC, LBUSY, SBUSY, LIDLE, SIDLE RANDOM, NRS(expr)			ORDER

Abb. A-2.3.1.2: GROUP-Block mit Parameterliste und Erläuterungen⁹⁴

Der GROUP-Block ermöglicht eine Gruppierung von Ressourcen. Zur Definition einer Ressourcengruppe sind in einem GROUP-Block der Gruppenname „GRLBL“, die Liste der zur Gruppe gehörenden Ressourcennummern „RESNUM“ und eine Ressourcenauswahlregel „GRSELRULE“ zu vereinbaren. Jedem GROUP-Block wird dabei von Visual SLAM (wie bei RESOURCE-Blöcken gehandhabt) ein numerischer Schlüssel (Nummer) zugeordnet, der benutzerspezifisch mit dem Parameter „GRNUM“ modifiziert werden kann⁹⁵.

Kapazitätsanforderungen an die einzelnen Ressourcen eines GROUP-Blocks werden wie bei RESOURCE-Blöcken mit Hilfe von AWAIT- oder PREEMPT-Knoten implementiert und entsprechende Ressourcenbelegungen können ebenso mit FREE-Knoten wieder freigegeben werden⁹⁶.

Mit der Ressourcen-Auswahlregel wird die Reihenfolge festgelegt, nach der die Ressourcenzuteilung für an einem AWAIT- bzw. PREEMPT-Knoten eintreffende Einheiten erfolgen soll. Die im GROUP-Block zur Verfügung stehenden Auswahlregeln werden in der nachfolgenden Tabelle wiedergegeben.

⁹⁴ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 347; vgl. hierzu Pritsker Corporation 1997 /Reference/ S. 36 f.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 186 f.

⁹⁵ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 186 f.; Pritsker Corporation 1997 /Reference/ S. 36 f.

⁹⁶ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 186 f. Hinsichtlich der Knoten AWAIT, PREEMPT und FREE sei verwiesen auf die entsprechenden Erläuterungen in (3) bis (5) in Abschnitt A-2.3.3.

Regelname	Definition
ORDER	Auswahl einer freien Ressource nach bevorzugter Reihenfolge
CYCLIC	Zyklische Auswahl der nächsten freien Ressource auf der Basis der zuletzt ausgewählten Ressource
LBUSY	Auswahl der Ressource mit der höchsten (aktuellen) Auslastung
SBUSY	Auswahl der Ressource mit der niedrigsten (aktuellen) Auslastung
LIDLE	Auswahl der Ressource mit der längsten Leerzeitdauer
SIDLE	Auswahl der Ressource mit der kürzesten Leerzeitdauer
RANDOM	Zufällige Auswahl aus freien Ressourcen nach vordefinierten Wahrscheinlichkeiten
NRS(expr)	Auswahl der Ressource gemäß benutzerdefinierter Routine. Der Wert von expr wird als ganzzahliger Code der Routine NRS ⁹⁷ im AWAIT-Knoten übertragen.

Tab. A-2.3.1.1: Ressourcen-Auswahlregeln innerhalb des GROUP-Blocks⁹⁸**(3) GATE-Block**

Netzwerksymbol:

NUM	GLBL	INITSTAT	IFL	repeats
-----	------	----------	-----	---------

Programmanweisung in Visual SLAM:

GATE,NUM,GLBL,INITSTAT,{IFL,repeats};

Eingabefeld	Option	Voreinstellung
NUM	positiver ganzzahliger konstanter Ausdruck	nächster sequentieller Index
GLBL	eindeutig identifizierbare Zeichenkette	Angabe erforderlich
INITSTAT	OPEN oder CLOSE	OPEN
IFL, repeats	positiver ganzzahliger konstanter Ausdruck	Angabe erforderlich

Abb. A-2.3.1.3: GATE-Block mit Parameterliste und Erläuterungen⁹⁹

Der GATE-Block dient zur Definition einer logischen Sperre, bezeichnet als GLBL, mit dem Anfangsstatus „OPEN“ oder „CLOSE“. Jedem GATE-Block wird von Visual SLAM (wie bei

⁹⁷ Vgl. zur Definition benutzerspezifischer Ressourcen-Auswahlregeln für GROUP-Blöcke mit Hilfe von Visual SLAM Programmerweiterungen in Visual Basic oder C Pritsker Corporation 1997 /Reference/ S. 81 f. u. 127 f.

⁹⁸ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 348; in Anlehnung an Pritsker, O'Reilly 1999 /Visual SLAM/ S. 186, Table 6-1.

⁹⁹ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 348; vgl. ferner Pritsker Corporation 1997 /Reference/ S. 34 f.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 209 f.

RESOURCE und GROUP-Blöcken) ein numerischer Schlüssel (Nummer) zugeordnet, der benutzerspezifisch mit dem Parameter „NUM“ modifiziert werden kann¹⁰⁰. Für einen GATE-Block ist mindestens eine Datei unter Angabe einer zugehörigen Dateinummer „IFL“ festzulegen, die als Warteraum für Einheiten dient, die mit Hilfe der Sperre aufgehalten werden sollen. Einheiten können mit Hilfe von AWAIT-Knoten, die auf eine Sperre nebst zugehöriger Warteraum- bzw. Warteschlangen-Datei referenzieren, aufgehalten werden, sofern die entsprechende Sperre den Status „CLOSE“ aufweist. Mit Hilfe des CLOSE- und des OPEN-Knotens kann der Anfangsstatus einer Sperre verändert werden¹⁰¹.

A-2.3.2 Knotenverbindungselemente

(1) SERVICE-Aktivität

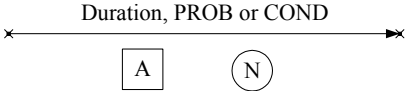
<u>Netzwerksymbol:</u>  <u>Programmanweisung in Visual SLAM:</u> ACTIVITY,A,Duration,PROB, or COND,NLBL,N,"ID";		
Eingabefeld	Option	Voreinstellung
A	positiver ganzzahliger beliebiger Ausdruck	unbesetzt, keine Statistik
Duration	beliebiger Ausdruck, REL(NLBL) oder STOPA(NTC)	0
PROB or COND	Wahrscheinlichkeit: PROB(expr), wobei der Wert des beliebigen Ausdrucks expr im Intervall [0;1] liegen sollte; Bedingung: Ausdruck	unbesetzt
NLBL unmittelbar	Zeichenkette	nächster Knoten, der der Aktivität nachfolgt
N	positiver ganzzahliger konstanter Ausdruck	1
ID	Zeichenkette	Leerzeichen

Abb. A-2.3.2.1: SERVICE-Aktivität mit Parameterliste und Erläuterungen¹⁰²

SERVICE-Aktivitäten weisen als Startknoten entweder einen QUEUE-Knoten oder einen SELECT-Knoten auf¹⁰³. Mit Hilfe des Parameters „A“ können Aktivitäten für die statistische Auswertung ihrer Nutzung von Einheiten (im Sinne des Einheitendurchlaufs) durchnumme-

¹⁰⁰ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 209; Pritsker Corporation 1997 /Reference/ S. 34 f.

¹⁰¹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 210 ff. Hinsichtlich des CLOSE- und des OPEN-Knotens sei verwiesen auf die entsprechenden Erläuterungen in (10) und (11) in Abschnitt A-2.3.3.

¹⁰² Entnommen aus Pütz 2004 /Produktionssysteme/ S. 343; vgl. hierzu ferner Pritsker Corporation 1997 /Reference/ S. 14; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 110 ff.

¹⁰³ Der QUEUE-Knoten wird in (13) in Abschnitt A-2.3.3 näher erläutert.

riert werden; die betreffende statistische Auswertung wird dann in einem Simulationsergebnisbericht (SUMMARY-Report) aufgeführt. SERVICE-Aktivitäten ermöglichen die Modellierung der Bedienung von Einheiten, beispielsweise stellvertretend für die Bearbeitung von Fertigungsaufträgen in einem realen zu modellierenden Produktionssystem. Mit SERVICE-Aktivitäten wird dabei ein Engpass abgebildet, da von den gleichzeitig an einer SERVICE-Aktivität eintreffenden Einheiten jeweils nur „N“ Einheiten von dieser Aktivität parallel bedient werden können; „N“ fungiert insofern als (Bedienungs-) Kapazität einer SERVICE-Aktivität¹⁰⁴. Der Wert von „N“, der bei der Modellierung einer Aktivität mit einer positiven ganzzahligen Konstante zu spezifizieren ist, kann während eines Simulationslaufs nicht modifiziert werden. Mit dem Parameter „Duration“ wird die Dauer der Verzögerung, der jeweils die Aktivität gleichzeitig durchlaufenden Einheiten (z.B. stellvertretend für die Rüstdauer eines Arbeitsgangs), vereinbart. Die Verzögerungsdauer kann ferner mit „REL(NLBL)“ in Abhängigkeit vom Zeitpunkt der nächsten Freigabe einer Einheit aus einem mit „NLBL“ bezeichneten Knoten vereinbart werden¹⁰⁵. Mit Hilfe der „STOPA(NTC)“-Spezifikation für „Duration“, wobei „NTC“ einen ganzzahligen beliebigen Ausdruck bezeichnet, dauert jeweils eine Verzögerung an, bis das „STOPA“ etwa mit Hilfe eines ASSIGN-Knotens¹⁰⁶ auf den Wert gem. „NTC“ gesetzt wird¹⁰⁷.

Haben mehrere SERVICE-Aktivitäten einen identischen Startknoten, so kann entweder mit Hilfe des Parameters „COND“ eine Bedingung oder mit Hilfe des Parameters „PROB“ eine Wahrscheinlichkeit als Kriterium zur Auswahl eines Zielknotens vereinbart werden¹⁰⁸, wodurch eine Steuerung der Weiterleitung von Einheiten ermöglicht wird. Anstelle einer direkten (graphischen) Verbindung eines Knotenpaares durch eine Aktivität kann diese Verbindung durch Spezifizierung von „NLBL“, den Zielknotenbezeichner der Aktivität¹⁰⁹, wie in der nachfolgenden Abbildung illustriert, indirekt implementiert werden.

¹⁰⁴ Vgl. hierzu die Erläuterung der Kapazität von SERVICE-Aktivitäten in SLAM II in Witte, Claus, Helling 1994 /SLAM/ S. 241.

¹⁰⁵ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 112.

¹⁰⁶ Vgl. zum ASSIGN-Knoten die Erläuterung in (2) in Abschnitt A-2.3.3.

¹⁰⁷ Vgl. zur „STOPA(NTC)“-Spezifikation Pritsker, O'Reilly 1999 /Visual SLAM/ S. 112., S. 526 u. S. 580.

¹⁰⁸ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 113 f.

¹⁰⁹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 111.

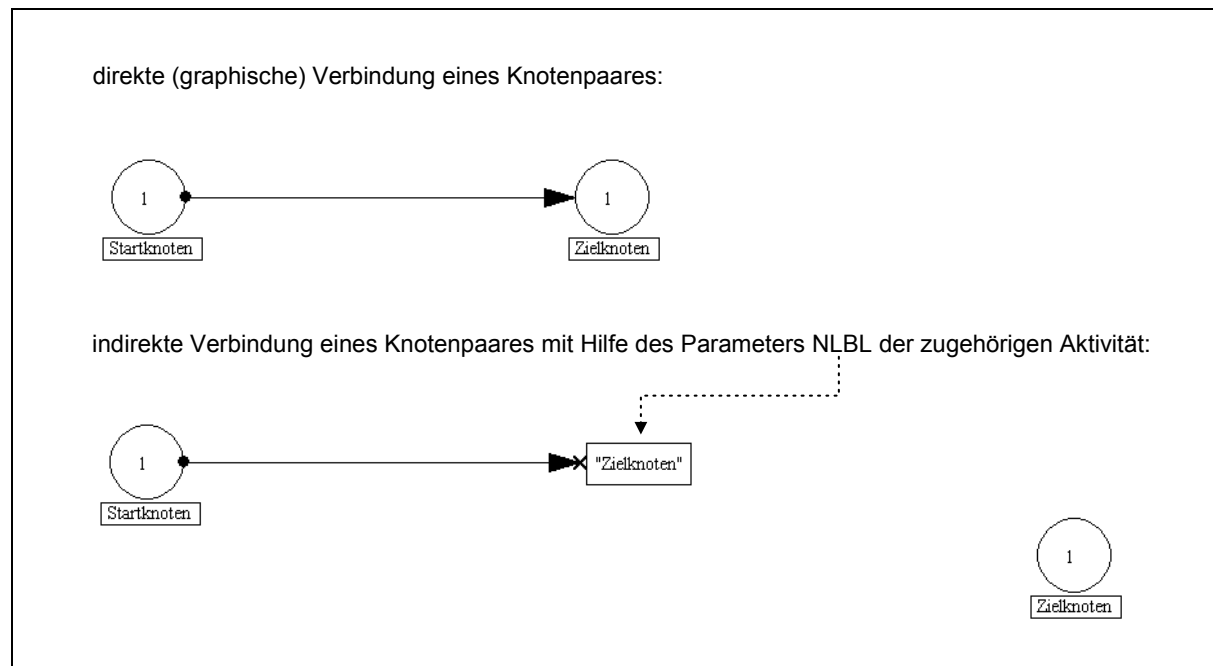


Abb. A-2.3.2.2: Möglichkeiten der Verbindung eines Knotenpaares¹¹⁰

(2) NON-SERVICE-Aktivität

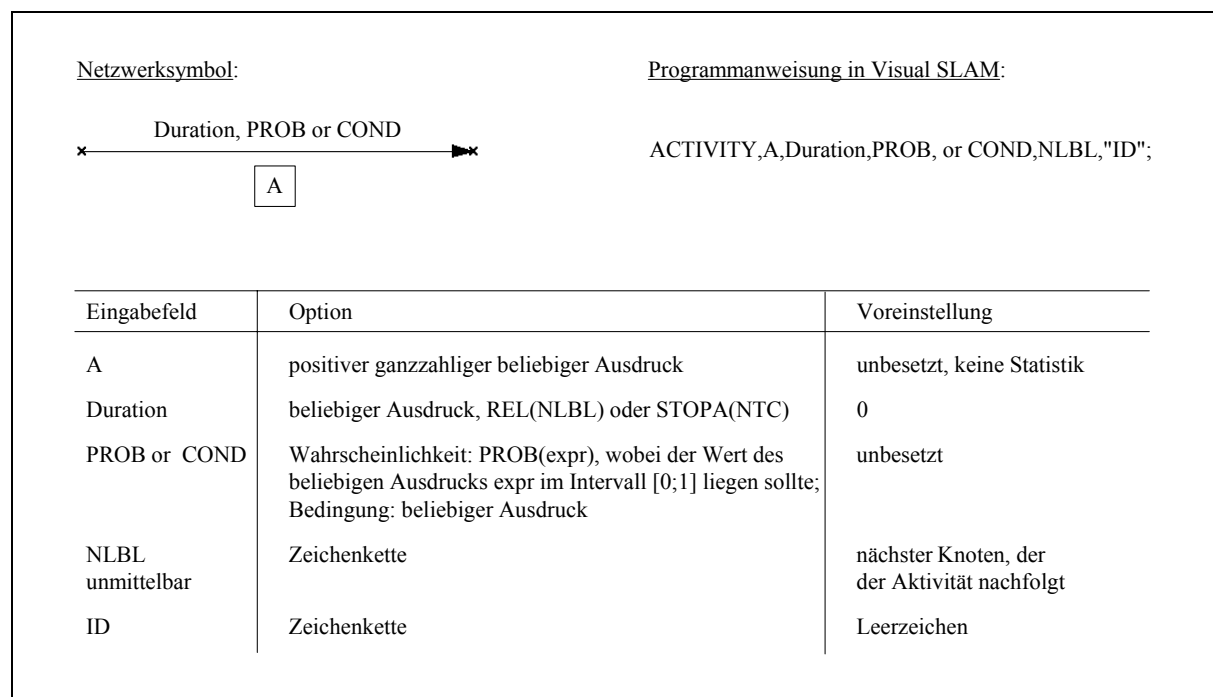


Abb. A-2.3.2.3: SERVICE-Aktivität mit Parameterliste und Erläuterungen¹¹¹

NON-SERVICE-Aktivitäten sind Aktivitäten, die als Startknoten einen anderen Knoten als einen SELECT- oder QUEUE-Knoten aufweisen. Im Gegensatz zur SERVICE-Aktivität wei-

¹¹⁰ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 344.

¹¹¹ Vgl. hierzu Pritsker Corporation 1997 /Reference/ S. 14; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 110 ff.

sen NON-SERVICE-Aktivitäten keine Beschränkung der Bedienungskapazität auf, d.h. beliebig viele, an NON-SERVICE Aktivitäten eintreffende Einheiten können von diesen Aktivitäten parallel bedient werden¹¹². Der Parameter „N“ der SERVICE-Aktivität ist deshalb auch nicht in der Parameterliste der NON-SERVICE-Aktivität enthalten. Ansonsten weisen die Parameterlisten beider Aktivitätenarten identische Parameter auf.

A-2.3.3 Knoten

Bei den Knoten in Visual SLAM, die einen zweiteiligen Aufbau, bestehend aus Funktions- und Weiterleitungsteil (mit dem Weiterleitungsparameter „M“), aufweisen, wird die Weiterleitung der jeweils an diesen Knoten eintreffenden Einheiten unter Berücksichtigung des Wertes des Weiterleitungsparameters „M“ (einem ganzzahligen beliebigen Visual SLAM-Ausdruck) und den vom zugrunde liegenden Knoten jeweils ausströmenden Aktivitäten nebst den für diese gegebenenfalls spezifizierten Bedingungen („COND“)¹¹³ bzw. Wahrscheinlichkeiten (Funktion „PROB“)¹¹⁴ vereinbart. Sofern die Anzahl der jeweils ausströmenden Aktivitäten größer als die durch den Wert des jeweiligen Weiterleitungsparameters festgelegte maximal zulässige Anzahl zu durchquerender Aktivitäten ist oder für die Aktivitäten Bedingungen bzw. Wahrscheinlichkeiten spezifiziert sind, erfolgt eine Auswahl der jeweils zu durchquerenden Aktivitäten. Dazu werden die Aktivitäten ausgehend von ihrem unmittelbaren graphischen Verlauf beim Austritt aus dem zugrunde liegenden Startknoten von oben nach unten nacheinander berücksichtigt. Erfolgt die Aktivitätenauswahl ausschließlich anhand von Bedingungen, so wird die jeweils weiterzuleitende Einheit zerstört, falls keine der implementierten Bedingungen erfüllt ist¹¹⁵. Aus den mit Wahrscheinlichkeiten spezifizierten Aktivitäten wird für jede den betreffenden Knoten verlassende Einheit von Visual SLAM jeweils maximal eine Aktivität ausgewählt. Dazu werden die einzelnen Wahrscheinlichkeiten der Platzierung der zugehörigen Aktivitäten von oben nach unten folgend aufkumuliert. Dabei wird der entsprechend „obersten“ Aktivität als kumulierte Wahrscheinlichkeit ihre in „PROB“ spezifizierte Wahrscheinlichkeit zugewiesen. Die nächsten nach unten folgenden Aktivitäten erhalten jeweils als kumulierte Wahrscheinlichkeit die Summe aus ihrer in „PROB“ spezifizierten Wahrscheinlichkeit und der kumulierten Wahrscheinlichkeit der unmittelbar über ihr platzierten Aktivität. Die Auswahlprozedur einer Aktivität wird sodann in Visual SLAM mit Hilfe einer im Intervall $[0, 1]$ gleichverteilten Zufallszahl, die mittels eines dafür reservierten

¹¹² Vgl. hierzu Pritsker Corporation 1997 /Reference/ S. 14; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 110 ff.

¹¹³ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 113 f.

¹¹⁴ Vgl. Pritsker Corporation 1997 /Reference/ S. 8; vgl. ferner Pritsker, O'Reilly 1999 /Visual SLAM/ S. 114 f.

¹¹⁵ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 114 f.

Zufallszahlenstroms generiert wird, erzeugt¹¹⁶. Ausgewählt wird nunmehr jene Aktivität, deren kumulierte Wahrscheinlichkeit entweder mit dem Wert der Zufallszahl übereinstimmt oder diesen mit kürzestem Abstand überschreitet. Für den Fall, dass die größte einer Aktivität zugeordnete kumulierte Wahrscheinlichkeit kleiner als die Zufallszahl ist, wird von Visual SLAM aus den mit Wahrscheinlichkeiten spezifizierten Aktivitäten keine ausgewählt. Betrifft dieser Fall einen Knoten, dessen ausströmende Aktivitäten ausnahmslos mit Wahrscheinlichkeiten spezifiziert sind, wird die Einheit, die den Auswahlprozess ausgelöst hat, zerstört. Zur Vermeidung dieses Falles ist es erforderlich, dass die Summe der mit Wahrscheinlichkeiten spezifizierten, knotenspezifisch ausströmenden Aktivitäten den Wert „1“ aufweist¹¹⁷ insbesondere, wenn mit der entsprechend modellierten zufallsbedingten Auswahl ein sicheres Ereignis im Simulationsmodell repräsentiert werden soll¹¹⁸.

Zum Verständnis der Auswahl der aus einem Knoten ausströmenden Aktivitäten, die zur Weiterleitung der den betreffenden Knoten verlassenden Einheiten spezifiziert sein können, wird nunmehr das in der nachfolgenden Abbildung angeführte Beispiel erläutert.

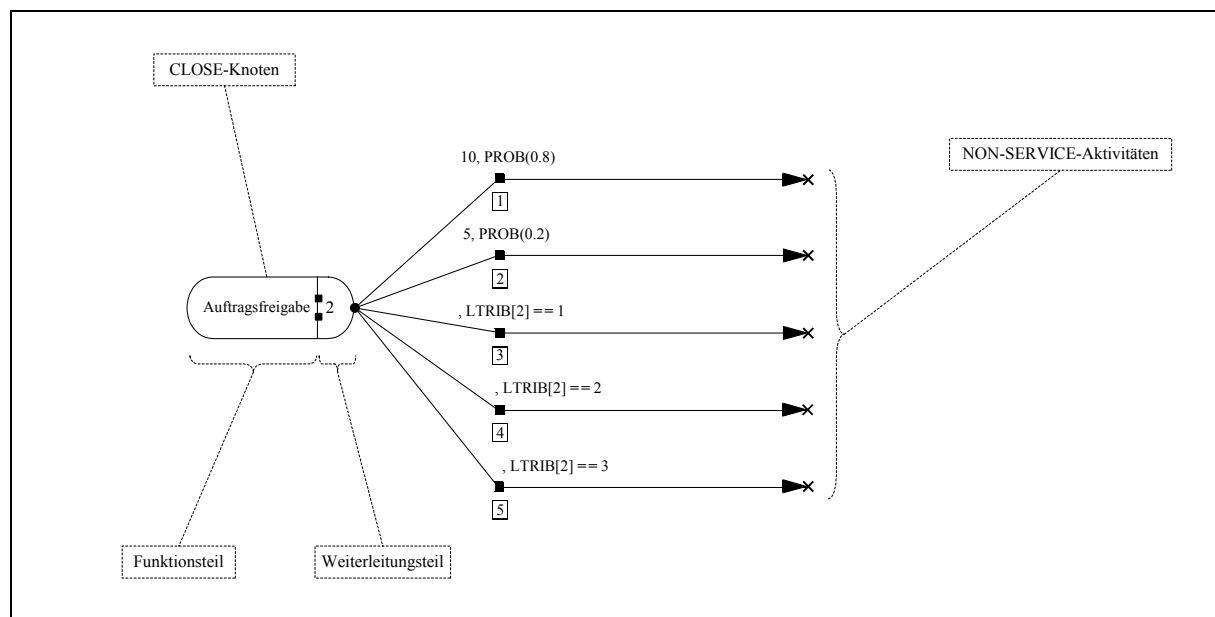


Abb. A-2.3.3.1: CLOSE-Knoten¹¹⁹ mit fünf ausströmenden NON-SERVICE-Aktivitäten¹²⁰

¹¹⁶ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 114; Pritsker Corporation 1997 /Reference/ S. 8.

¹¹⁷ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 91 f.

¹¹⁸ Vgl. zum Gegenstand des sicheren Ereignisses innerhalb der Wahrscheinlichkeitstheorie z.B. Dillmann 1990 /Statistik I/ S. 51.

¹¹⁹ Der CLOSE-Knoten wird innerhalb des aktuellen Abschnittes weiter unten in (10) erläutert.

¹²⁰ Diese Abbildung ist entnommen aus Pütz 2004 /Produktionssysteme/ S. 351.

Gemäß $M = 2$ können von den im obigen Beispiel modellierten Aktivitäten maximal 2 zur Weiterleitung der den Knoten jeweils verlassenden Einheit dienen. Von den mit Wahrscheinlichkeiten spezifizierten Aktivitäten (Nr. 1 und Nr. 2) wird von Visual SLAM wie weiter oben erläutert, eine Aktivität ausgewählt. Je nach dem Variableninhalt der einheitenspezifischen Variablen „LTRIB[2]“, kann ferner eine der mit Bedingungen spezifizierten Aktivitäten mit den Nummern 3 bis 5 zur Weiterleitung herangezogen werden, so dass gegebenenfalls die den Knoten verlassende Einheit von Visual SLAM einmal kopiert wird.

(1) CREATE-Knoten

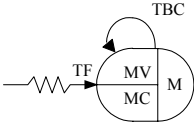
<u>Netzwerksymbol:</u>		<u>Programmanweisung in Visual SLAM:</u>
		CREATE,TBC,TF,MV,MC,M;
Eingabefeld	Option	Voreinstellung
TBC	positiver globaler Ausdruck	unendlich
TF	positiver globaler Ausdruck	0
MV	linksseitiger einheitenspezifischer Ausdruck	unbesetzt
MC	positiver ganzzahliger globaler Ausdruck	unendlich
M	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.3.3.2: CREATE-Knoten mit Parameterliste und Erläuterungen¹²¹

Mit dem CREATE-Knoten werden im Abstand von „TBC“-Zeiteinheiten Einheiten generiert. Mit „TF“ wird der Generierungszeitpunkt der ersten Einheit (des entsprechenden CREATE-Knotens) spezifiziert. Der Parameter „MC“ dient zur Beschränkung der Anzahl zu generierender Einheiten. Sofern der Parameter „MV“ als Variable spezifiziert wird, werden die Generierungszeitpunkte der zu erzeugenden Einheiten von Visual SLAM in dieser Variablen gespeichert¹²². Mit dem Parameter „M“ wird, wie bereits oben angeführt, die maximale Anzahl der von dem betreffenden Knoten ausströmenden Aktivitäten spezifiziert, die von einer den Knoten verlassenden Einheit durchquert werden können.

¹²¹ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 352; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 26 f.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 99 f.

¹²² Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 99.

(2) ASSIGN-Knoten

Netzwerksymbol: VAR = VALUE repeats M Programmanweisung in Visual SLAM: ASSIGN, {{VAR,VALUE},repeats},M;		
Eingabefeld	Option	Voreinstellung
VAR	globale oder einheitenspezifische Visual SLAM Variable	Angabe erforderlich
VALUE	beliebiger Ausdruck (Datentyp von VALUE und zugehörigem VAR müssen übereinstimmen.)	Angabe erforderlich
M	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.3.3.3: ASSIGN-Knoten mit Parameterliste und Erläuterungen¹²³

Mit dem ASSIGN-Knoten werden Variablen Werte bzw. Inhalte zugewiesen. ASSIGN-Knoten können dabei jeweils eine Liste entsprechender (Variablen-) Wertzuweisungen beinhalten, wobei in jeder Listenzeile jeweils eine Wertzuweisung von rechts nach links spezifiziert werden kann¹²⁴.

¹²³ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 352; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 17.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 107 ff.

¹²⁴ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 108.

(3) AWAIT-Knoten

<u>Netzwerksymbol:</u> 		<u>Programmanweisung in Visual SLAM:</u> AWAIT,IFL,{ {RESORGATE,UR} repeats,RULE,QC,FULLCOND,M;
Eingabefeld	Option	Voreinstellung
IFL	positiver ganzzahliger beliebiger Ausdruck	erste Dateinummer in der Dateiliste von RESORGATE
RESORGATE	Zeichenkette oder ganzzahliger beliebiger Ausdruck, welche(r) auf RESOURCE oder GATE referenziert.	Angabe erforderlich
UR	positiver ganzzahliger beliebiger Ausdruck (falls RESORGATE auf GATE referenziert, ohne Bedeutung)	1
RULE	ALL, ONE(expr) oder ALLOC(I)	ALL
QC	nicht-negativer ganzzahliger konstanter Ausdruck	unbegrenzt
FULLCOND	NONE, BLOCK oder BALK(NLBL)	NONE
M	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.3.3.4: AWAIT-Knoten mit Parameterliste und Erläuterungen¹²⁵

AWAIT-Knoten werden jeweils in Verbindung mit einem Visual SLAM Block (RESOURCE, GROUP oder GATE) verwendet¹²⁶.

(3a) AWAIT-Knoten in Verbindung mit RESOURCE- bzw. GROUP-Blöcken:

Im AWAIT-Knoten wird „RESORGATE“ als Ressource (bzw. Ressourcengruppe) spezifiziert. Am AWAIT-Knoten ankommende Einheiten fordern von den dort als „RESORGATE“ spezifizierten Ressourcen /-gruppen jeweils „UR“ (Kapazitäts-) Einheiten an. Reichen die aktuell zu einem Simulationszeitpunkt verfügbaren Kapazitäten nicht aus, um die entsprechenden Anforderungen zu erfüllen, werden die anfordernden Einheiten in der Warteschlangendatei „IFL“ mit der Kapazität „QC“ verzögert, bis die verfügbaren Kapazitäten die zugehörigen angeforderten „UR“-Werte erreicht haben. Ist RESORGATE als Ressource spezifiziert, so muss die Dateinummer „IFL“ in der zugehörigen Ressourcendateiliste aufgeführt sein¹²⁷. Ist demgegenüber RESORGATE als Ressourcengruppe spezifiziert, so muss die Dateinummer „IFL“ in den Ressourcendateilisten sämtlicher zur entsprechenden Ressourcen-

¹²⁵ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 353; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 18 f.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 166 ff.

¹²⁶ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 167.

¹²⁷ Vgl. ebenda S. 167.

gruppe gehörenden Ressourcen enthalten sein¹²⁸. Mit „RULE“ wird eine Regel spezifiziert, nach der die angeforderten Ressourcen aus der Ressourcenliste des AWAIT-Knotens zugeteilt werden. Die verfügbaren Regeln sind in der nachfolgenden Tabelle aufgeführt.

Regelname	Definition
ALL	Eine eintreffende Einheit wird so lange verzögert, bis alle im AWAIT-Knoten aufgelisteten Einheiten verfügbar sind.
ONE(expr)	Die erste, innerhalb der Liste verfügbare Ressource wird zugeteilt. Die Nummer der demzufolge zugeteilten Ressource wird im ganzzahligen linksseitigen einheitenspezifischen Ausdruck expr gespeichert.
ALLOC(I)	Die Ressourcenzuteilung erfolgt mit Hilfe der benutzerdefinierten Funktion ALLOC ¹²⁹ mit I als ganzzahligem Eingabeparameter.

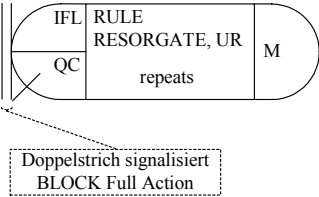
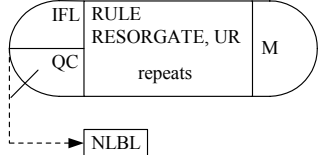
Tab. A-2.3.3.1: Ressourcen-Zuteilungsregeln innerhalb des AWAIT-Knotens in Visual SLAM¹³⁰

Ist eine Maximalkapazität „QC“ der Warteschlange „IFL“ spezifiziert worden, so kann für den Fall der Überschreitung von „QC“ gemäß „FULLCOND“ eine sog. „Full Action“, wie in der nachfolgenden Tabelle angeführt, ausgelöst werden.

¹²⁸ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 186.

¹²⁹ Vgl. zur Implementierung benutzerspezifisierter Ressourcen-Zuteilungsregeln mit Hilfe von Visual SLAM Programmerweiterungen in Visual Basic oder C Pritsker, O'Reilly 1999 /Visual SLAM/ S. 512 ff. u. S. 565 ff.; Pritsker Corporation 1997 /Reference/ S. 74 f. u. S. 120 f.

¹³⁰ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 354; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 167 ff.; Pritsker Corporation 1997 /Reference/ S. 19.

FULLCOND	Definition
NONE	Am AWAIT-Knoten eintreffende Einheiten werden zerstört.
BLOCK	- Bei NON-SERVICE-Aktivität als dem AWAIT-Knoten vorausgehende Aktivität: am AWAIT-Knoten eintreffende Einheiten werden zerstört. - Bei SERVICE-Aktivität als dem AWAIT-Knoten vorausgehende Aktivität: Einheiten werden auf dieser SERVICE-Aktivität blockiert, bis die Maximalkapazität von QC unterschritten wird. AWAIT-Knoten mit BLOCK Full Action: 
BALK(NLBL)	Am AWAIT-Knoten eintreffende Einheiten werden (ausweichend) zum Knoten NLBL gemäß BALK(NLBL) geleitet. AWAIT-Knoten mit BALK Full Action: 

Tab. A-2.3.3.2: FULLCOND mit zugehörigen 'Full Actions' innerhalb des AWAIT-Knotens in Visual SLAM¹³¹

(3b) AWAIT-Knoten in Verbindung mit GATE-Blöcken:

Im AWAIT-Knoten wird „RESORGATE“ als „GATE“ (logische Sperre) spezifiziert, wobei der zugehörige Parameter „UR“ keine Bedeutung hat. Einheiten, die am AWAIT-Knoten eintreffen, werden so lange in der Warteschlangendatei „IFL“ verzögert, wie der Status der Sperre RESORGATE geschlossen ist. Alle verzögerten Einheiten werden wieder freigesetzt, wenn der Status der Sperre auf 'offen' gesetzt wird¹³². Abgesehen von „UR“ ist die Bedeutung der einzelnen Parameter des AWAIT-Knotens unabhängig von der Spezifizierung des Parameters „RESORGATE“.

¹³¹ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 355; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 18 ff.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 167 u. S. 169.

¹³² Vgl. Pritsker Corporation 1997 /Reference/ S. 18 f.

(4) PREEMPT-Knoten

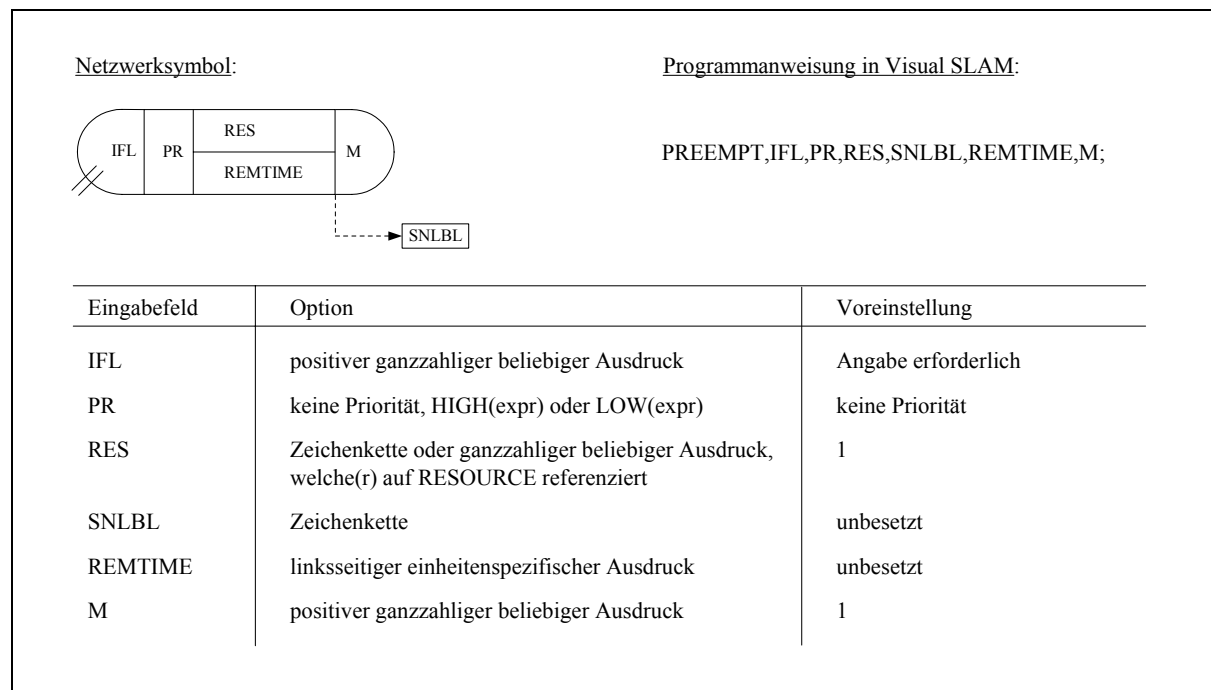


Abb. A-2.3.3.5: PREEMPT-Knoten mit Parameterliste und Erläuterungen¹³³

Wie bereits oben hinsichtlich des AWAIT-Knotens dargestellt, dient auch der PREEMPT-Knoten zur Belegung einer Ressourcenkapazität, wobei von PREEMPT-Knoten stets nur eine Kapazitätseinheit der betreffenden Ressource angefordert werden kann. Ressourcenbelegungen, die mit Hilfe von AWAIT-Knoten vereinbart wurden, werden i.d.R. durch Ressourcenanforderungen, die mit PREEMPT-Knoten vereinbart werden, unmittelbar verdrängt. Diese Verdrängung bleibt aus, sofern die Verzögerungsdauer „Duration“ in der dem AWAIT-Knoten folgenden NON-SERVICE Aktivität mit „REL(NLBL)“ oder „STOPA(NTC)“¹³⁴ spezifiziert ist¹³⁵. Einheiten, die sich nicht in NON-SERVICE Aktivitäten befinden, können nicht verdrängt werden¹³⁶. Ist die Kapazität einer zugrunde liegenden Ressource „RES“ größer als 1 und sind alle zugehörigen Kapazitätseinheiten belegt, so kann mit Hilfe eines PREEMPT-Knotens nur die jeweils aktuell zuletzt belegte Kapazitätseinheit verdrängt werden¹³⁷.

Konkurrieren mehrere PREEMPT-Knoten um die Belegung einer Ressource, dann erfolgt die Ressourcenbelegung in Abhängigkeit von der jeweils im PREEMPT-Knoten spezifizierten

¹³³ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 356; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 39 f.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 194 f.

¹³⁴ Vgl. zur REL(NLBL) und STOPA(NTC) Spezifikation die Erläuterungen in Abschnitt A-2.3.2 unter (1).

¹³⁵ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 194.

¹³⁶ Vgl. Pritsker Corporation 1997 /Reference/ S. 39.

¹³⁷ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 194.

Priorität „PR“. Diese Priorisierung kann jeweils bezogen auf eine bestimmte Ressource „RES“ entweder mit Hilfe der Spezifizierung „HIGH(expr)“ oder „LOW(expr)“ vereinbart werden, wobei expr einen beliebigen Ausdruck bezeichnet¹³⁸. Dabei verdrängen bei „HIGH(expr)“ [bzw. „LOW(expr)“] Ressourcenanforderungen mit höheren (bzw. niedrigeren) Variablenwerten (gemäß expr) Ressourcenanforderungen mit niedrigeren (bzw. höheren) Variablenwerten. Kommt es dabei zu keiner Verdrängung, werden die Einheiten, die an einem PREEMPT-Knoten mit niedriger Priorität eintreffen, in der zugehörigen Warteschlangendatei „IFL“ verzögert. Verdrängte Einheiten können durch Spezifikation eines Zielknotenbezeichners „SNLBL“ zum Knoten mit dem spezifizierten Bezeichner (SNLBL) geleitet werden. Die infolge einer Verdrängung noch ausstehende Restbedienungszeit kann in einer Variablen, die mit „REMTIME“ spezifiziert wird, gespeichert werden¹³⁹. Ist „SNLBL“ nicht spezifiziert, wird die jeweils von einem PREEMPT-Knoten verdrängte Einheit von Visual SLAM in einer Unterbrechungsdatei („interrupt file“) als die erste, auf die verdrängungsspezifische Ressource wartende Einheit zwischengespeichert. Sobald eine Ressource wieder einer verdrängten Einheit zugewiesen wird, erfolgt die Wiederaufnahme ihrer Bedienung gemäß der noch ausstehenden Restbedienungszeit¹⁴⁰, selbst wenn diese nicht in „REMTIME“ gespeichert wurde.

¹³⁸ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 194.

¹³⁹ Vgl. ebenda S. 195.

¹⁴⁰ Vgl. ebenda S. 195.

(5) FREE-Knoten

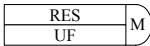
<u>Netzwerksymbol:</u>		<u>Programmanweisung in Visual SLAM:</u>
		FREE, {{RES,UF},repeats},M;
Eingabefeld	Option	Voreinstellung
RES	Zeichenkette oder positiver ganzzahliger beliebiger Ausdruck, der auf eine Ressource referenziert.	Angabe erforderlich
UF	positiver ganzzahliger beliebiger Ausdruck	1
M	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.3.3.6: FREE-Knoten mit Parameterliste und Erläuterungen¹⁴¹

Die jeweils an einem FREE-Knoten eintreffenden Einheiten führen dazu, dass „UF“ (Kapazitäts-) Einheiten der mit RES spezifizierten Ressource freigesetzt werden. „UF“ kann als ganzzahlige Konstante bzw. ganzzahliger beliebiger Ausdruck vereinbart werden. In FREE-Knoten kann jeweils eine Liste von Freisetzungen zugehöriger Kapazitätseinheiten implementiert werden¹⁴², wobei pro Listenzeile jeweils eine derartige Freisetzung spezifiziert werden kann. Wird „RES“ nicht spezifiziert, werden alle Ressourceneinheiten freigesetzt, die der (die Ressourcen freisetzenden) Einheit zugeteilt worden sind¹⁴³. Sofern Einheiten in den Warteschlangen der freigesetzten Ressourcen auf Ressourcenzuteilungen warten, werden diese Anforderungen von Visual SLAM, sofern möglich, erfüllt¹⁴⁴.

¹⁴¹ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 357; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 33 f.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 170 ff.

¹⁴² Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 170.

¹⁴³ Vgl. ebenda S. 171.

¹⁴⁴ Vgl. Pritsker Corporation 1997 /Reference/ S. 33; Pritsker, O'Reilly 1997 /Reference/ S. 170.

(6) ALTER-Knoten

<u>Netzwerksymbol:</u>		<u>Programmanweisung in Visual SLAM:</u>
		ALTER,RES,CC,M;
Eingabefeld	Option	Voreinstellung
RES	Zeichenkette oder positiver ganzzahliger beliebiger Ausdruck, der auf eine Ressource referenziert.	Angabe erforderlich
CC	positiver ganzzahliger beliebiger Ausdruck	Angabe erforderlich
M	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.3.3.7: ALTER-Knoten mit Parameterliste und Erläuterungen¹⁴⁵

Der ALTER-Knoten dient jeweils zur Veränderung (Erhöhung oder Reduzierung) der Kapazität der Ressource „RES“ um den Wert des positiven ganzzahligen beliebigen Ausdrucks „CC“. Falls die gemäß „CC“ zu reduzierende Ressourcenkapazität belegt ist, erfolgt die Reduzierung erst zum Simulationszeitpunkt der Freigabe der entsprechenden Ressourcenkapazität. Kapazitätsreduzierungen werden in Visual SLAM nur bis auf den Wert „0“ (Kapazitätseinheiten) unterstützt, weitergehende Reduzierungsanweisungen werden ignoriert¹⁴⁶.

¹⁴⁵ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 358; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 16; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 178 f.

¹⁴⁶ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 179.

(7) GOON-Knoten

<u>Netzwerksymbol:</u>		<u>Programmanweisung in Visual SLAM:</u>
		GOON,M;
Eingabefeld	Option	Voreinstellung
M	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.3.3.8: GOON-Knoten mit Parameterliste und Erläuterungen¹⁴⁷

GOON-Knoten weisen keinen speziellen Funktionsteil auf, sondern jeweils nur den Weiterleitungsparameter „M“ und dienen daher lediglich als Weiterleitungs- bzw. Fortsetzungsknoten, den alle jeweils dort eintreffenden Einheiten unmittelbar durchlaufen¹⁴⁸.

(8) BATCH-Knoten

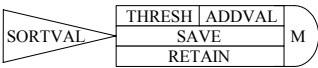
<u>Netzwerksymbol:</u>		<u>Programmanweisung in Visual SLAM:</u>
		BATCH,SORTVAL,THRESH,ADDVAL,SAVE{SUM,repeats},RETAIN,M;
Eingabefeld	Option	Voreinstellung
SORTVAL	beliebiger Ausdruck	0 (keine Sortierung)
THRESH	beliebiger Ausdruck	Angabe erforderlich
ADDVAL	positiver beliebiger Ausdruck	1
SAVE	FIRST, LAST, LOW(expr), HIGH(expr) oder USER(expr)	LAST
SUM	einheitenspezifischer linksseitiger Ausdruck	unbesetzt
RETAIN	YES oder NO	NO
M	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.3.3.9: BATCH-Knoten mit Parameterliste und Erläuterungen¹⁴⁹

¹⁴⁷ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 358; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 35 f.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 122.

¹⁴⁸ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 122.

¹⁴⁹ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 359; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 21 f.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 248 ff.

BATCH-Knoten dienen zur Gruppierung von Einheiten zu Mengeneinheiten. Von einem BATCH-Knoten wird jeweils eine solche Mengeneinheit freigesetzt, wenn die Summe der „ADDVAL“-Werte der gesammelten bzw. gruppierten Einheiten einer Menge den Schwellenwert „THRESH“ erreicht. Wird die Voreinstellung von „ADDVAL“ (ADDVAL = 1) beibehalten, so entspricht „THRESH“ der Anzahl der Einheiten, die eine Einheitengruppe enthalten soll. „THRESH“ kann als beliebiger Ausdruck spezifiziert werden und wird jeweils, beim Eintreffen der ersten Einheit einer zu gruppierenden Menge, entsprechend zugewiesen. Sofern „ADDVAL“ als einheitenspezifischer linksseitiger Ausdruck vereinbart ist, kann beim Eintreffen einer Einheit mit negativem „ADDVAL“-Wert unabhängig vom Erreichen des Schwellenwertes eine Mengeneinheit freigesetzt werden¹⁵⁰. Wird die Sortierungsgröße „SORTVAL“ von der Voreinstellung abweichend spezifiziert, können multiple (Teil-) Mengen gruppiert werden, wobei eine Zuordnung der Einheiten gemäß „SORTVAL“ erfolgt. Falls dabei eine am BATCH-Knoten eintreffende Einheit eine Ausprägung von „SORTVAL“ aufweist, für die noch keine Teilmenge besteht, wird eine neue Teilmenge angelegt¹⁵¹.

Mit dem „SAVE“-Kriterium wird festgelegt, welche Eigenschaften die zu bildenden Mengeneinheiten erhalten sollen¹⁵². Die „SAVE“-Optionen des „SAVE“-Kriteriums sind in der nachfolgenden Tabelle aufgeführt.

„SAVE“-Option	Definition
FIRST	Die Mengeneinheit enthält die Eigenschaften der Einheit, die als erste in die Gruppe eingegliedert wird.
LAST	Die Mengeneinheit enthält die Eigenschaften der Einheit, die als letzte in die Gruppe eingegliedert wird.
LOW(expr)	Die Mengeneinheit enthält die Eigenschaften der Einheit mit dem niedrigsten Wert des beliebigen Ausdrucks „expr“.
HIGH(expr)	Die Mengeneinheit enthält die Eigenschaften der Einheit mit dem höchsten Wert des beliebigen Ausdrucks „expr“.
USER(expr)	Die Mengeneinheit enthält die Eigenschaften eines benutzerdefinierten Kriteriums ¹⁵³ gemäß „expr“.

Tab. A-2.3.3.3: „SAVE“-Optionen des „SAVE“-Kriteriums innerhalb des BATCH-Knotens in Visual SLAM¹⁵⁴

¹⁵⁰ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 248.

¹⁵¹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 248 f.; Pritsker Corporation 1997 /Reference/ S. 21.

¹⁵² Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 249.

¹⁵³ Zur Implementation benutzerspezifisierter SAVE-Kriterien dient die BATCHSAVE-Funktion für Visual SLAM Programmiererweiterungen in Visual Basic oder C. Vgl. hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 526 u. 580; Pritsker Corporation 1997 /Reference/ S. 77 u. S. 123.

¹⁵⁴ Weitgehend entnommen aus Pütz 2004 /Produktionssysteme/ S. 360; vgl. ferner Pritsker, O'Reilly 1999 /Visual SLAM/ S. 249; Pritsker Corporation 1997 /Reference/ S. 21.

Jede dieser in Tabelle A-2.3.3.2 aufgelisteten Optionen des SAVE-Kriteriums lässt sich mit Hilfe einer Liste von Attributen (einheitenspezifische Variablen) modifizieren, deren einheitenspezifische Werte durch Spezifikation von SUM aufsummiert werden. Die vom BATCH-Knoten freizugebenden (Teil-) Mengeneinheiten erhalten dabei für die in der „SUM“ Spezifikation aufgelisteten Attribute jeweils die korrespondierenden aufsummierten Werte als Attributwerte¹⁵⁵.

Die einzelnen Einheiten, die mit Hilfe der BATCH-Knoten zu (Teil-) Mengeneinheiten gruppiert werden, können mit Hilfe von UNBATCH-Knoten¹⁵⁶ freigesetzt werden. Mit dem Parameter „RETAIN“ wird dabei für einen späteren Zugriff auf die einzelnen Einheiten einer (Teil-) Mengeneinheit bestimmt, ob die Eigenschaften der einzelnen Einheiten gespeichert werden sollen („YES“-Spezifikation) oder nicht („NO“-Spezifikation)¹⁵⁷.

(9) UNBATCH-Knoten

Netzwerksymbol:		Programmanweisung in Visual SLAM:
		UNBATCH,NCLONE,M;
Eingabefeld	Option	Voreinstellung
NCLONE	positiver ganzzahliger beliebiger Ausdruck	unbesetzt
M	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.3.3.10: UNBATCH-Knoten mit Parameterliste und Erläuterungen¹⁵⁸

Der UNBATCH-Knoten dient einerseits dazu, um mit BATCH-Knoten jeweils zu Mengeneinheiten gruppierte Einheiten (wieder) freizusetzen und andererseits aus eintreffenden Einheiten „NCLONE“ Klone zu generieren. Wird die Voreinstellung von „NCLONE“ zugrundegelegt, werden im Falle des Eintreffens einer mit einem BATCH-Knoten generierten Men-

¹⁵⁵ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 249 f.

¹⁵⁶ Siehe hierzu die nachfolgenden Erläuterungen in (9).

¹⁵⁷ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 250. In Visual SLAM steht zur Gruppierung von Einheiten ferner der ACCUMULATE-Knoten zur Verfügung, der im Gegensatz zum BATCH-Knoten keine temporäre, sondern eine endgültige Gruppierung von Einheiten veranlasst; hinsichtlich des ACCUMULATE-Knotens sei verwiesen auf Pritsker, O'Reilly 1999 /Visual SLAM/ S. 240 ff.; Pritsker Corporation 1997 /Reference/ S. 12 f.

¹⁵⁸ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 361; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 51; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 252.

geneinheit die zugehörigen Einheiten freigesetzt; anderenfalls wird die eintreffende Einheit unmittelbar unter Berücksichtigung der „M“-Spezifikation weitergeleitet¹⁵⁹. Falls „NCLONE“ abweichend von der Voreinstellung spezifiziert ist, wird diese Spezifikation ignoriert, falls eine gruppierte Einheit am BATCH-Knoten eintrifft¹⁶⁰. In diesem Fall werden lediglich die zugehörigen Einheiten freigesetzt. Wird dabei „NCLONE“ mit dem Wert „1“ parametrisiert, so werden eintreffende nicht-gruppierte Einheiten wiederum unmittelbar unter Berücksichtigung der M-Spezifikation weitergeleitet¹⁶¹, da eine derartige eintreffende Einheit bereits der Anzahl „1“ zu erzeugender Klone entspricht.

(10) CLOSE-Knoten

<u>Netzwerksymbol:</u>		<u>Programmanweisung in Visual SLAM:</u>
		CLOSE,GATE,M;
Eingabefeld	Option	Voreinstellung
GATE	Zeichenkette oder ganzzahliger beliebiger Ausdruck, der auf ein GATE referenziert.	1
M	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.3.3.11: CLOSE-Knoten mit Parameterliste und Erläuterungen¹⁶²

Die am CLOSE-Knoten eintreffenden Einheiten lösen jeweils das Schließen der dort mit „GATE“ spezifizierten logischen Sperre aus¹⁶³.

¹⁵⁹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 252.

¹⁶⁰ Vgl. ebenda S. 252.

¹⁶¹ Vgl. ebenda S. 252.

¹⁶² Entnommen aus Pütz 2004 /Produktionssysteme/ S. 362; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 24; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 210.

¹⁶³ Vgl. Pritsker Corporation 1997 /Reference/ S. 24.

(11) OPEN-Knoten

<u>Netzwerksymbol:</u>		<u>Programmanweisung in Visual SLAM:</u>
		OPEN,GATE,M;
Eingabefeld	Option	Voreinstellung
GATE	Zeichenkette oder ganzzahliger beliebiger Ausdruck, der auf ein GATE referenziert.	1
M	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.3.3.12: OPEN-Knoten mit Parameterliste und Erläuterungen¹⁶⁴

Analog zur Funktionalität des CLOSE-Knotens dienen die jeweils an einem OPEN-Knoten eintreffenden Einheiten zum Öffnen der dort mit „GATE“ spezifizierten logischen Sperre¹⁶⁵.

(12) TERMINATE-Knoten

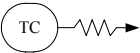
<u>Netzwerksymbol:</u>		<u>Programmanweisung in Visual SLAM:</u>
		TERMINATE,TC;
Eingabefeld	Option	Voreinstellung
TC	positiver ganzzahliger beliebiger Ausdruck	unendlich

Abb. A-2.3.3.13: TERMINATE-Knoten mit Parameterliste und Erläuterungen¹⁶⁶

Mithilfe des TERMINATE-Knotens werden die dort jeweils ankommenden Einheiten zerstört bzw. gelöscht. Wird im betreffenden TERMINATE-Knoten der Zerstörungszähler „TC“ spezifiziert, dient dieser Knoten zudem zur Steuerung des Simulationslaufs. Dies manifestiert

¹⁶⁴ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 362; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 38 f.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 210.

¹⁶⁵ Vgl. Pritsker Corporation 1997 /Reference/ S. 38 f.

¹⁶⁶ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 363; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 50 f.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 106.

sich darin, dass mit der Zerstörung der „TC“-ten am entsprechenden Knoten eintreffenden Einheit der aktuell zugrunde liegende Simulationslauf beendet wird¹⁶⁷.

(13) QUEUE-Knoten

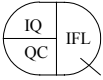
<u>Netzwerksymbol:</u>		<u>Programmanweisung in Visual SLAM:</u>
		QUEUE,IFL,IQ,QC,FULLCOND,{SLBL,repeat};
Eingabefeld	Option	Voreinstellung
IFL	positiver ganzzahliger beliebiger Ausdruck	Angabe erforderlich
IQ	nicht-negativer ganzzahliger konstanter Ausdruck	0
QC	nicht-negativer ganzzahliger konstanter Ausdruck	unendlich
FULLCOND	NONE, BLOCK oder BALK(NLBL)	NONE
SLBL,repeat: Knotenbezeichner der dem QUEUE-Knoten unmittelbar folgenden SELECT- oder MATCH-Knoten; wird von Visual SLAM anhand der zugehörigen Netzwerkdaten generiert.		

Abb. A-2.3.3.14: QUEUE-Knoten mit Parameterliste und Erläuterungen¹⁶⁸

Der QUEUE-Knoten dient jeweils zum Einfügen von Einheiten in eine Warteschlangendatei „IFL“¹⁶⁹. Durch Spezifikation von „IFL“ als positiven ganzzahligen beliebigen Ausdruck kann eine Warteschlangendateiauswahl implementiert werden. Existiert dabei während eines Simulationslaufs zum aktuellen „IFL“-Wert noch keine Warteschlangendatei, dann wird diese beim nächsten Eintreffen einer Einheit am entsprechenden QUEUE-Knoten von Visual SLAM angelegt¹⁷⁰. Durch Spezifikation von „IQ“ können vor Beginn eines Simulationslaufes Einheiten in die Warteschlangendatei „IFL“ eingestellt werden¹⁷¹.

Ist eine Maximalkapazität „QC“ der Warteschlange „IFL“ spezifiziert worden, so kann, wie beim AWAIT-Knoten möglich¹⁷², für den Fall einer Überschreitung von „QC“ gemäß „FULLCOND“ auch in Verbindung mit dem jeweiligen QUEUE-Knoten eine „Full Action“

¹⁶⁷ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 106.

¹⁶⁸ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 363; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 41 f.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 102 ff.

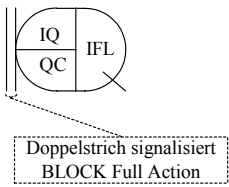
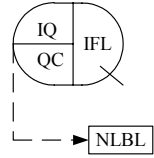
¹⁶⁹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 102.

¹⁷⁰ Vgl. Pritsker Corporation 1997 /Reference/ S. 41.

¹⁷¹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 102; Pritsker Corporation 1997 /Reference S. 41.

¹⁷² Siehe hierzu entsprechend die Erläuterungen zu „FULLCOND“ beim AWAIT-Knoten in (3) im aktuellen Abschnitt.

ausgelöst werden¹⁷³. Die „FULLCOND“-Optionen sind in der nachfolgenden Tabelle aufgelistet:

FULLCOND	Definition
NONE	Am QUEUE-Knoten eintreffende Einheiten werden zerstört.
BLOCK	- Bei NON-SERVICE-Aktivität als dem QUEUE-Knoten vorausgehende Aktivität: am QUEUE-Knoten eintreffende Einheiten werden zerstört. - Bei SERVICE-Aktivität als dem QUEUE-Knoten vorausgehende Aktivität: Einheiten werden auf dieser SERVICE-Aktivität blockiert bis die Maximalkapazität von QC unterschritten wird. QUEUE-Knoten mit BLOCK Full Action:  Doppelstrich signalisiert BLOCK Full Action
BALK(NLBL)	Am QUEUE-Knoten eintreffende Einheiten werden (ausweichend) zum Knoten mit dem Knotenbezeichner NLBL gemäß BALK(NLBL) geleitet. QUEUE-Knoten mit BALK Full Action: 

Tab. A-2.3.3.4: FULLCOND mit zugehörigen „Full Actions“ innerhalb des QUEUE-Knotens in Visual SLAM¹⁷⁴

Bei den jeweils aus einem QUEUE-Knoten ausströmenden Aktivitäten handelt es sich stets um eine SERVICE-Aktivität. Da QUEUE-Knoten keinen spezifizierbaren Weiterleitungsparameter „M“ aufweisen, wird bei mehreren vom betreffenden QUEUE-Knoten ausströmenden SERVICE-Aktivitäten aufgrund der entsprechenden „PROB“- oder „COND“-Spezifikationen¹⁷⁵ maximal eine Aktivität ausgewählt. Als den QUEUE-Knoten unmittelbar folgende Netzwerkelemente sind in Visual SLAM neben den SERVICE-Aktivitäten lediglich

¹⁷³ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 103 f.; Pritsker Corporation 1997 /Reference/ S. 41 f.

¹⁷⁴ Weitestgehend entnommen aus Pütz 2004 /Produktionssysteme/ S. 364; vgl. ferner Pritsker Corporation 1997 /Reference/ S. 41 ff.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 103 f.

¹⁷⁵ Vgl. zur „PROB“- und „COND“-Spezifikation innerhalb von SERVICE-Aktivitäten die Ausführungen in (1) in Abschnitt A-2.3.2.

MATCH-Knoten und SELECT-Knoten, die jeweils mittels Konnektoren¹⁷⁶ mit einem QUEUE-Knoten zu verbinden sind¹⁷⁷, möglich.

(14) MATCH-Knoten

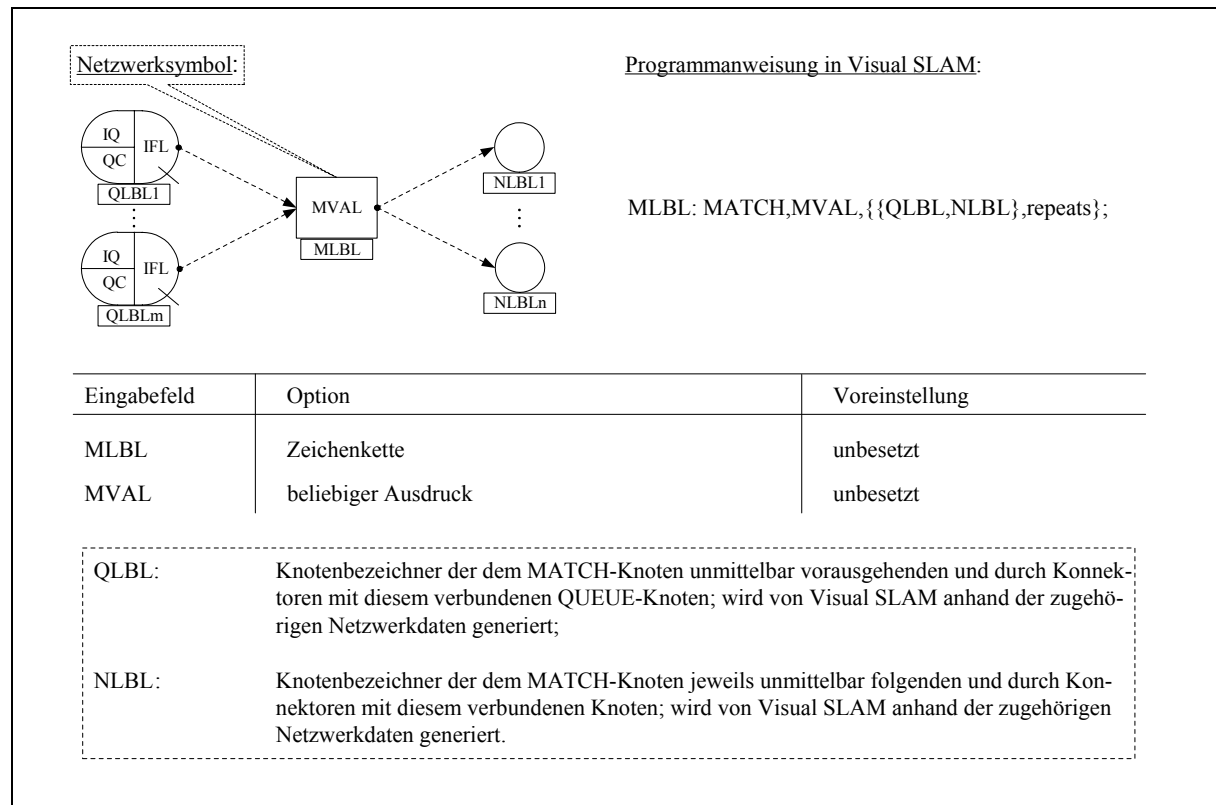


Abb. A-2.3.3.15: MATCH-Knoten mit Parameterliste und Erläuterungen¹⁷⁸

Der MATCH-Knoten dient jeweils zur Synchronisation von Einheiten in Warteschlangendateien „QLBL“, die unmittelbar einem solchen MATCH-Knoten vorausgehen und mit diesem durch Konnektoren verbunden sind. Für jede Warteschlangendatei „QLBL“, deren Einheiten mit Hilfe des MATCH-Knotens synchronisiert weitergeleitet werden sollen, wird ein entsprechender Knoten „NLBL“ (als Zielknoten) mit dem MATCH-Knoten durch einen Konnektor verbunden¹⁷⁹. Die Weiterleitungs-Zuordnung von „QLBL“ und „NLBL“ wird aufgrund der im jeweiligen Netzwerk eingezeichneten Konnektoren von oben nach unten in Visual SLAM generiert. Sobald in jeder der mit dem MATCH-Knoten verbundenen Warteschlangendateien „QLBL“ eine Einheit enthalten ist und diese Einheiten hinsichtlich des Synchronisationsaus-

¹⁷⁶ Vgl. hierzu die entsprechenden Ausführungen in Abschnitt 3.2.2.2.1.2 in (bb) unter (3).

¹⁷⁷ Vgl. hierzu die Ausführungen zum QUEUE-Knoten in SLAM II in Witte, Claus, Helling 1994 /SLAM/ S. 271; vgl. ferner Pritsker Corporation 1997 /Reference/ S. 41.

¹⁷⁸ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 365; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 37 f.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 263 ff.

¹⁷⁹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 263.

drucks „MVAL“ eine Entsprechung aufweisen, findet eine Synchronisation statt¹⁸⁰, die mit der Weiterleitung der entsprechenden Einheiten gemäß der vereinbarten Weiterleitungs-Zuordnung abgeschlossen wird. Dabei werden jene Einheiten, für deren Warteschlangendatei keine Weiterleitungszuordnung spezifiziert ist, zerstört¹⁸¹.

Die Anwendung des MATCH-Knotens unterliegt der Beschränkung, dass als dem MATCH-Knoten unmittelbar vorausgehende Knoten lediglich QUEUE-Knoten zulässig sind und ein MATCH-Knoten mit anderen Knoten grundsätzlich nur durch Konnektoren verbunden werden kann¹⁸².

(15) COLCT-Knoten

<u>Netzwerksymbol:</u>		<u>Programmanweisung in Visual SLAM:</u>
		COLCT,N,VALUE,ID,NCEL,HLOW,HWID,M;
Eingabefeld	Option	Voreinstellung
N	ganzzahliger beliebiger Ausdruck	nächster sequentieller Index
VALUE	beliebiger Ausdruck	Angabe erforderlich
ID	Zeichenkette	unbesetzt
NCEL	positiver ganzzahliger konstanter Ausdruck	unbesetzt (kein Histogramm)
HLOW	positiver konstanter globaler Ausdruck	unbesetzt
HWID	positiver konstanter globaler Ausdruck	unbesetzt
M	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.3.3.16: COLCT-Knoten mit Parameterliste und Erläuterungen¹⁸³

COLCT-Knoten dienen zur Erfassung des (jeweils) beim Eintreffen einer Einheit an einem derartigen Knoten aktuell vorhandenen Wertes des dort spezifizierten beliebigen Ausdrucks „VALUE“. Die betreffenden Werte werden in der „COLCT“-Statistiktable des zugehörigen Szenarios gespeichert. In Visual SLAM stehen zur Spezifizierung von VALUE zwei eigens für den COLCT-Knoten definierte globale reelle Variablen, und zwar „FIRSTARRIVE“ und

¹⁸⁰ Vgl. Witte, Claus, Helling 1994 /SLAM/ S. 266.

¹⁸¹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 263.

¹⁸² Vgl. Pritsker O'Reilly 1999 /Visual SLAM/ 263; Witte, Claus, Helling 1994 /SLAM/ S. 267.

¹⁸³ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 366; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 25 f.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 125 ff.

„LASTARRIVE, zur Verfügung¹⁸⁴. Letztere werden in der nachfolgenden Tabelle angeführt, in der eine Auswahl gemeinhin üblicher Spezifizierungen von „VALUE“ aufgelistet ist.

„VALUE“-Spezifikation	Definition
FIRSTARRIVE	Die Ankunftszeit der ersten am COLCT-Knoten eintreffenden Einheit wird gespeichert.
TNOW	Die Ankunftszeit sämtlicher am COLCT-Knoten eintreffender Einheiten wird gespeichert.
TNOW – LASTARRIVE	Die Zwischenankunftszeit (d.h. die Zeit, die zwischen zwei am COLCT-Knoten eintreffenden Einheiten anfällt) der eintreffenden Einheiten wird gespeichert. Als Referenzzeitpunkt wird dabei die Ankunftszeit der ersten Einheit gewählt.
TNOW – ATRIB[I]	Die Ankunftszeit abzüglich des Wertes der reellen einheitenspezifischen Variablen ATRIB[I] jeder ankommenden Einheit wird gespeichert.

Tab. A-2.3.3.5: Ausgewählte, gemeinhin übliche Spezifizierungen des VALUE-Ausdrucks innerhalb des COLCT-Knotens in Visual SLAM¹⁸⁵

Neben dieser Auswahl kann VALUE grundsätzlich mit einem beliebigen Ausdruck, der auch die benutzerspezifische Funktion USERF¹⁸⁶ beinhalten kann, parametrisiert werden¹⁸⁷.

Die mit Hilfe eines COLCT-Knotens generierten (statistischen) Daten können in einem Histogramm dargestellt werden. Hierzu sind jeweils die Klassenanzahl „NCEL“, die untere Grenze der ersten Klasse „HLOW“ und die Klassenbreite „HWID“ zu vereinbaren¹⁸⁸. Für jeden COLCT-Knoten werden in der Rubrik „OBSERVED STATISTICS REPORT“ des Summary Reports einleitend mit der COLCT Bezeichnung gemäß „ID“ Durchschnitt, Standardabweichung, minimaler und maximaler Wert sowie die Anzahl der erfassten Daten angegeben¹⁸⁹.

¹⁸⁴ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 125.

¹⁸⁵ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 367; vgl. ferner Pritsker, O'Reilly 1999 /Visual SLAM/ S. 125; Pritsker Corporation 1997 /Reference/ S. 25.

¹⁸⁶ Vgl. zur Implementierung der benutzerspezifisierbaren Funktion USERF in Visual SLAM Programmerweiterungen in Visual Basic oder C Pritsker, O'Reilly 1999 /Visual SLAM/ S. 500 ff. u. 551 ff.; Pritsker Corporation 1997 /Reference/ S. 87 f. u. S. 134 f. sowie die Erläuterungen in Abschnitt 4.2.1.5.3.1.

¹⁸⁷ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 125.

¹⁸⁸ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 125; Pritsker Corporation 1997 /Reference/ S. 25 f.

¹⁸⁹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 125; Pritsker Corporation 1997 /Reference/ S. 25 f.

(16) DETECT-Knoten

Netzwerksymbol:

VALUE	XDIR	THRESH	TOL	M
-------	------	--------	-----	---

Programmanweisung in Visual SLAM:

DETECT,VALUE,XDIR,THRESH,TOL,M;

Eingabefeld	Option	Voreinstellung
VALUE	globaler Ausdruck	Angabe erforderlich
XDIR	POSITIVE, NEGATIVE oder EITHER	EITHER
THRESH	globaler Ausdruck	Angabe erforderlich
TOL	positiver konstanter Ausdruck	unendlich
M	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.3.3.17: DETECT-Knoten mit Parameterliste und Erläuterungen¹⁹⁰

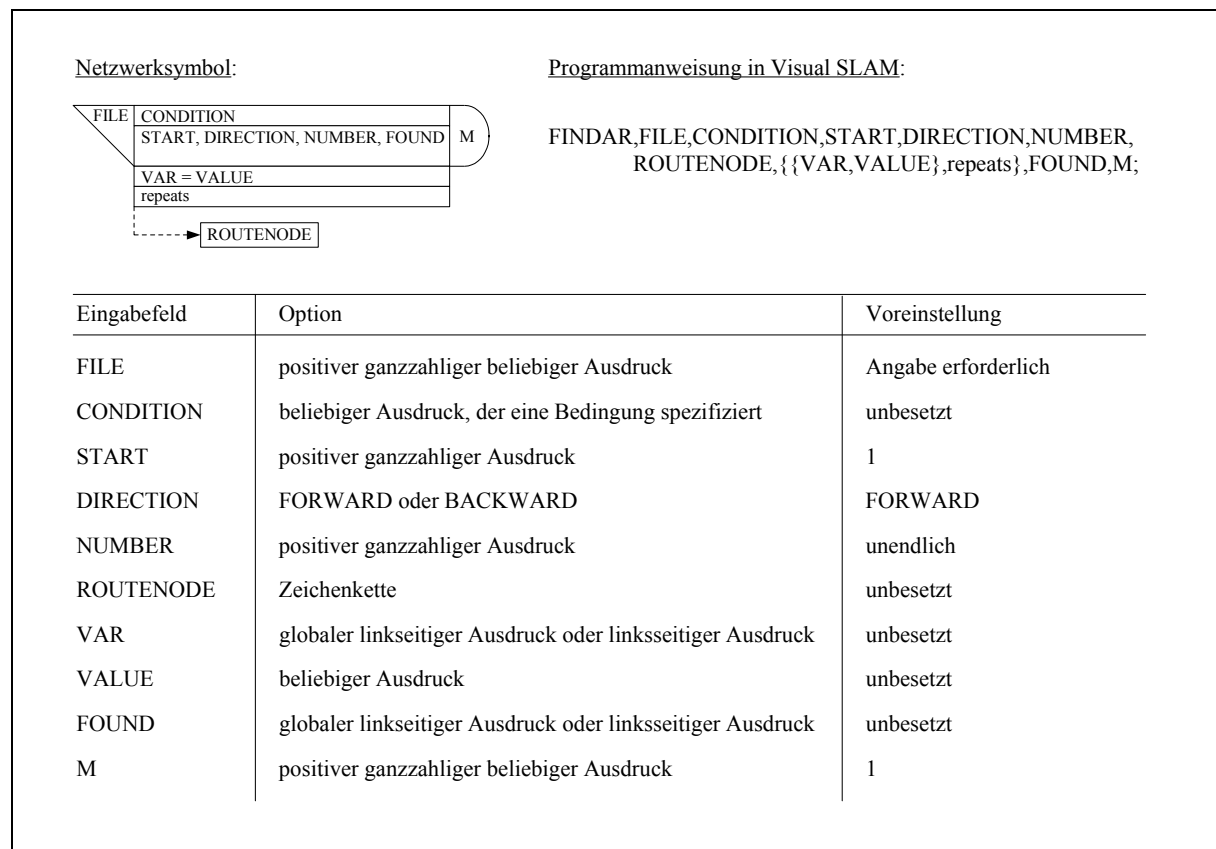
Der DETECT-Knoten ermöglicht jeweils eine Generierung von Einheiten in Abhängigkeit vom Über- oder auch Unterschreiten (bzw. Kreuzen) eines Schwellenwertes „THRESH“ bezogen auf einen mit „VALUE“ zu spezifizierenden Wert. Dabei wird mit „XDIR“ die Richtung des Kreuzens des Schwellenwertes spezifiziert¹⁹¹. Wird der DETECT-Knoten innerhalb eines kontinuierlichen Simulationsmodells bzw. -modellausschnitts verwendet, kann für kontinuierliche Variablen ein Toleranzwert für die Abweichung vom Schwellenwert in Form einer positiven Konstanten vereinbart werden¹⁹². Hierdurch wird jeweils die Generierung einer Einheit veranlasst, sobald der Wert von VALUE innerhalb dieser Toleranz liegt. Einheiten, die an einem DETECT-Knoten eintreffen, werden unmittelbar weitergeleitet¹⁹³.

¹⁹⁰ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 368; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 28.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 231 u. S. 599.

¹⁹¹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 231.

¹⁹² Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 231 u. S. 599 ff.

¹⁹³ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 231.

(17) FINDAR-KnotenAbb. A-2.3.3.18: FINDAR-Knoten mit Parameterliste und Erläuterungen¹⁹⁴

Der FINDAR-Knoten dient einerseits jeweils zur Suche von Einheiten, die sich in der Warteschlangendatei „FILE“ befinden und für die die in „CONDITION“ spezifizierte Bedingung erfüllt ist. Mit dem Parameter „START“ wird dabei das Glied (bzw. der Einheitenplatz) der Warteschlangendatei spezifiziert, mit dem die Suche nach den entsprechenden Einheiten begonnen werden soll. Mit „DIRECTION“ wird die Suchrichtung vorgegeben. Mit dem Parameter „NUMBER“ wird die Anzahl der Einheiten spezifiziert, für welche die Bedingung „CONDITION“ erfüllt sein soll. Wird diese Anzahl während eines Suchvorgangs erreicht, wird der Suchvorgang beendet, andernfalls werden sämtliche Einheiten der Warteschlangendatei überprüft. Die Anzahl der nach Abschluss eines Suchvorgangs bedingungsspezifisch aufgefundenen Einheiten kann in der in „FOUND“ zu spezifizierenden Variablen gespeichert werden¹⁹⁵. Für die aufgefundenen Einheiten kann mit Ausnahme des Rangfolgeattributes ferner eine Liste von Variablenwertzuweisungen gemäß der Spezifikation der Variablen in

¹⁹⁴ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 369; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 32; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 233 f.

¹⁹⁵ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 233.

„VAR“ und dem jeweils zuzuweisenden Wert in „VALUE“ veranlasst werden¹⁹⁶. Die aufgefundenen Einheiten können ferner zu einem (Ziel-)Knoten weitergeleitet werden, sofern dessen Bezeichner im Parameter „ROUTENODE“ des FINDAR-Knotens spezifiziert ist¹⁹⁷.

Der FINDAR-Knoten dient andererseits - sofern „CONDITION“ nicht spezifiziert ist - dazu, mit Hilfe von „VAR“ und „VALUE“ zu spezifizierende Variablenwertzuweisungen zu veranlassen. Diese Wertzuweisungen werden für sämtliche Einheiten der Warteschlangendatei vorgenommen, falls für „NUMBER“ die Voreinstellung akzeptiert wird. Anderenfalls wird mit „NUMBER“ die Anzahl der Einheiten begrenzt, wobei eine Auswahl der Einheiten unter Berücksichtigung der Parameter „START“ und „DIRECTION“ veranlasst wird¹⁹⁸. Die entsprechenden Einheiten werden ferner bei Spezifikation von „ROUTENODE“ zu dem entsprechend bezeichneten (Ziel-) Knoten weitergeleitet¹⁹⁹.

Die am FINDAR-Knoten eintreffenden Einheiten lösen jeweils die dort spezifizierten Vorgänge aus und verlassen daraufhin unmittelbar diesen Knoten unter Berücksichtigung des im FINDAR-Knoten spezifizierten M-Parameters und der zugehörigen ausströmenden Aktivitäten.

¹⁹⁶ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 233.

¹⁹⁷ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 233.

¹⁹⁸ Vgl. ähnlich hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 234.

¹⁹⁹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 234.

(18) READ-Knoten

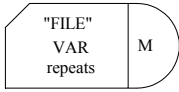
<u>Netzwerksymbol:</u>		<u>Programmanweisung in Visual SLAM:</u>
		READ,"FILE",REOPEN,STORVAR,"FORMAT",{VAR,repeats},M;
Eingabefeld	Option	Voreinstellung
"FILE"	Zeichenkette (Dateibezeichner)	Angabe erforderlich
REOPEN	YES oder NO	NO
STORVAR	Variable	unbesetzt
"FORMAT"	scanf-Ausdruck	unbesetzt (Zeichenfolge abgegrenzt durch Leerzeichen)
VAR, repeats	Variable	Angabe erforderlich
M	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.3.3.19: READ-Knoten mit Parameterliste und Erläuterungen²⁰⁰

Der READ-Knoten ermöglicht jeweils das Einlesen von Daten aus einer externen Datei in ein Visual SLAM Netzwerk. Die am READ-Knoten eintreffenden Einheiten veranlassen jeweils das Einlesen einer Zeichenfolge bzw. eines Datensatzes. Im entsprechenden READ-Knoten ist hierzu der Dateiname „FILE“ der auszulesenden Datei zu spezifizieren. Die Umwandlung der Inhalte der Zeichenfolge können entsprechend der Funktion „scanf“²⁰¹ der Programmiersprache C durch Spezifikation von „FORMAT“ vereinbart werden.

Als generelle Formatspezifikation mit optionalen Größen in eckigen Klammern wird dabei %[Feldbreite]Format-Kodierung oder \Format-Kodierung verwendet²⁰². Die einzelnen Format-Kodierungen des READ-Knotens, die auch im weiter unten angeführten WRITE-Knoten Anwendung finden, werden in der nachfolgenden Tabelle dargestellt.

²⁰⁰ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 370; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 42 ff.; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 223 f.

²⁰¹ Vgl. zur Verwendung der scanf-Funktion der Programmiersprache C Kernighan, Richie 1990 /C/ S. 150 ff. u. S. 243 ff.

²⁰² Vgl. ähnlich hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 222 f.

Format-Kodierung	Beschreibung
%c	Zeichen
%s	Zeichenkette
%d	dezimaler ganzzahliger Wert mit Vorzeichen
%ld	langer ganzzahliger Wert mit Vorzeichen
%u	ganzzahliger Wert ohne Vorzeichen
%f	Gleitpunktwert (reeller Wert)
%lf	langer doppelt genauer Gleitpunktwert
\b	Rückwärtsschritt
\f	Seitenvorschub
\n	Zeilentrenner
\r	Wagenrücklauf
\t	Tabulatorzeichen

Tab. A-2.3.3.6: Format-Kodierungen des READ- und WRITE-Knotens in Visual SLAM²⁰³

Die Variablen zur Speicherung der eingelesenen Daten werden in „VAR“ spezifiziert. Wird „FORMAT“ nicht spezifiziert, wird in Visual SLAM ein durch Leerzeichen begrenztes Standardformat für die einzulesenden Variablenwerte zugrunde gelegt²⁰⁴.

Mit der ersten am READ-Knoten eintreffenden Einheit wird die auszulesende Datei geöffnet. Werden mehrere aufeinander folgende Simulationsläufe innerhalb eines Szenarios realisiert, kann mit der „YES“-Option des Parameters „REOPEN“ veranlasst werden, dass die auszulesende Datei bei jedem Simulationslauf erneut geöffnet und das Einlesen wiederholt wird²⁰⁵. In „STORVAR“ kann eine Variable spezifiziert werden, in der jeweils die Anzahl der bei einem Lesevorgang aktuell eingelesenen Werte gespeichert wird. Sofern „STORVAR“ nicht spezifiziert ist, löst ein unvollständiger Lesevorgang einen Ausführungsfehler aus²⁰⁶.

²⁰³ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 371 in Anlehnung an Pritsker, O'Reilly 1999 /Visual SLAM/ S. 223, Table 7-1.

²⁰⁴ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 223.

²⁰⁵ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 223.

²⁰⁶ Vgl. Pritsker Corporation 1997 /Reference/ S. 43.

(19) WRITE-Knoten

<u>Netzwerksymbol:</u>		<u>Programmanweisung in Visual SLAM:</u>
		WRITE,"FILE",FLUSH,"FORMAT",{VALUE,repeats},M;
Eingabefeld	Option	Voreinstellung
"FILE"	Zeichenkette (Dateibezeichner)	Angabe erforderlich
FLUSH	YES oder NO	NO
"FORMAT"	printf-Ausdruck	unbesetzt
VALUE, repeats	Ausgabewert(e)	Angabe erforderlich
M	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.3.3.20: WRITE-Knoten mit Parameterliste und Erläuterungen²⁰⁷

Der WRITE-Knoten dient jeweils dazu, mit „VALUE“ spezifizierte Werte in eine externe Datei mit der Bezeichnung „FILE“ zu schreiben. Hierzu werden die Werte entsprechend der „printf“-Funktion der Programmiersprache C durch Spezifikation von „FORMAT“ in Zeichenfolgen umgewandelt²⁰⁸. Als generelle Formatspezifikation mit optionalen Größen in eckigen Klammern wird dabei „%[Feldbreite][.][Stellen nach Dezimalpunkt]Format-Kodierung“ oder „\Format-Kodierung“ verwendet²⁰⁹. Die zu dieser Umwandlung zur Verfügung stehenden Format-Kodierungen stimmen mit denjenigen des READ-Knotens überein²¹⁰. Die Ausgabedatei wird beim ersten Eintreffen einer Einheit am entsprechenden WRITE-Knoten erzeugt. Jede jeweils an einem WRITE-Knoten eintreffende Einheit löst einen dort spezifizierten Schreib- bzw. Ausgabevorgang aus. Falls diese jeweils einzeln in der externen Datei ausgeführt werden sollen, ist der Parameter „FLUSH“ mit der Option „YES“ zu spezifizieren²¹¹. Durch die Voreinstellung „NO“ (des Parameters „FLUSH“) werden die ausgelösten Schreibvorgänge während eines Simulationslaufs aufgezeichnet und nach dessen Beendigung in der externen Datei ausgeführt. Hierdurch wird pro Simulationslauf jeweils nur eine Verbindung zu der entsprechenden externen Datei aufgebaut, wodurch die Schreibvorgänge schneller als

²⁰⁷ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 372; vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 53; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 225 f.

²⁰⁸ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 225.

²⁰⁹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 222 f.

²¹⁰ Vgl. hierzu Tabelle 4.2.10.

²¹¹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 225.

mit der Option „YES“ realisiert werden können. Diesem Geschwindigkeitsvorteil steht der Nachteil gegenüber, dass die während eines Simulationslaufs aufgezeichneten Schreibvorgänge bei einem lauffeilerbedingten Abbruch eines Simulationslaufs verloren gehen²¹².

A-2.3.4 Funktionen

(1) Liste sämtlicher globaler Funktionen in Visual SLAM

Funktion	Bedeutung
ASSERT(VAL, LOW, HIGH)	Diese Funktion überprüft, ob der Wert „VAL“ im Bereich zwischen dem Wert „LOW“ und dem Wert „HIGH“ liegt. Als Ergebnis wird der Wert Value geliefert. Falls Value außerhalb des Bereichs liegt, wird ein Fehler generiert.
GETEXCEL(SHEET, IROW, ICOL)	Diese Funktion dient dazu, die Daten aus der durch die Zeile „IROW“ und Spalte „ICOL“ definierten Zelle der Excel-Tabelle mit dem Dateinamen „SHEET“ in ein (Sub-) Netzwerk einzulesen ²¹³ .
GGTBLN(IRX, IRY, XVALUE)	Diese Funktion liefert den Wert von „ARRAY[IRY, Position]“, wobei für „Position“ die zum Wert „XVALUE“ zugehörige Position des ARRAY „IRX“ gewählt wird.
NGUSE(I)	Diese Funktion liefert die aktuelle Anzahl belegter Kapazitätseinheiten der Ressourcengruppe „I“.
NNACT(I)	Diese Funktion liefert die Anzahl der Einheiten, die von der Aktivität „I“ aktuell bedient werden.
NNCNT(I)	Diese Funktion liefert die Anzahl der Einheiten, die von der Aktivität „I“ bedient worden sind.
NNGAT(I)	Diese Funktion liefert den aktuellen Status der Sperre „I“.
NNGRP(I)	Diese Funktion liefert die aktuelle Anzahl freier Kapazitätseinheiten der Ressourcengruppe „I“.
NNRSC(I)	Diese Funktion liefert die aktuelle Anzahl freier Kapazitätseinheiten der Ressource „I“.
NRUSE(I)	Diese Funktion liefert die aktuelle Anzahl belegter Kapazitätseinheiten der Ressource „I“.
NNQ(I)	Diese Funktion liefert die Anzahl der Einheiten, die aktuell in der (Warteschlangen-) Datei „I“ sind.
PROB(P)	Diese Funktion liefert den Wert WAHR, falls eine Zufallszahl im Intervall [0,1] kleiner als „P“ ist, wobei „P“ im Intervall [0,1] liegen muss. Diese Funktion ist nur in der Spezifikation des Parameters „COND“ in Aktivitäten zulässig.
USERF(N)	Diese Funktion liefert den Wert der benutzerdefinierten Funktion mit der Nummer „N“.
VSNEPR(VSN, INST, EXPR)	Diese Funktion liefert den Wert des Ausdrucks „EXPR“ der Instanz „INST“ des objektorientierten Subnetzwerkes (kurz OSN) „VSN“.

Tab. A-2.3.4.1: Liste sämtlicher globaler Funktionen in Visual SLAM²¹⁴

²¹² Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 225.

²¹³ Die zugrunde liegende Excel-Datei muss als „Microsoft Excel 4.0 Tabelle“ implementiert sein, damit der Exceldateiname identisch ist mit dem Namen der Exceltabelle; vgl. Pütz 2004 /Produktionssysteme/ S. 373.

²¹⁴ Weitestgehend entnommen aus Pütz 2004 /Produktionssysteme/ S. 601 in Anlehnung an Pritsker Corporation 1997 /Reference/ S. 6, Table 2.5.

(2) Liste sämtlicher Zufallsgenerator-Funktionen in Visual SLAM

Funktion	Bedeutung
DRAND(IS)	Diese Funktion liefert eine auf Basis des Zufallszahlenstroms „IS“ ²¹⁵ generierte und im Intervall [0, 1] gleichverteilte Zufallszahl.
EXPON(XMN,IS)	Diese Funktion liefert eine mit dem Mittelwert „XMS“ auf Basis des Zufallszahlenstroms „IS“ generierte exponentialverteilte Zufallszahl.
UNFRM(ULO,UHI,IS)	Diese Funktion liefert eine im Intervall [ULO, UHI] auf Basis des Zufallszahlenstroms „IS“ generierte gleichverteilte Zufallszahl.
WEIBL(ALPHA,BETA,IS)	Diese Funktion liefert eine mit den Parametern „ALPHA“ und „BETA“ auf Basis des Zufallszahlenstroms „IS“ generierte weibullverteilte Zufallszahl.
TRIAG(XLO,XMODE,XHI,IS)	Diese Funktion liefert eine im Intervall [XLO, XHI] mit dem Moduswert „XMODE“ dreiecksverteilte Zufallszahl auf Basis des Zufallszahlenstroms „IS“.
RNORM(XMN,STD,IS)	Diese Funktion liefert eine mit dem Mittelwert „XMN“ und der Standardabweichung „STD“ auf Basis des Zufallszahlenstroms „IS“ generierte normalverteilte Zufallszahl.
RLOGN(XMN,STD,IS)	Diese Funktion liefert eine mit „XMN“ und „STD“ auf Basis des Zufallszahlenstroms „IS“ generierte logarithmisch normalverteilte Zufallszahl.
ERLNG(EMN,XK,IS)	Diese Funktion liefert eine erlangverteilte Zufallszahl, die aus der Summe von „XK“ Exponentialverteilungen und dem Mittelwert „EMN“ auf Basis des Zufallszahlenstroms „IS“ generiert wird.
GAMA(BETA,ALPHA,IS)	Diese Funktion liefert eine mit den Parametern „BETA“ und „ALPHA“ auf Basis des Zufallszahlenstroms „IS“ generierte gammavereilte Zufallszahl.
BETA(THETA,PHI,IS)	Diese Funktion liefert eine mit den Parametern „THETA“ und „PHI“ auf Basis des Zufallszahlenstroms „IS“ generierte betaverteilte Zufallszahl.
NPSSN(XMN,IS)	Diese Funktion liefert eine mit dem Mittelwert „XMN“ auf Basis des Zufallszahlenstroms „IS“ generierte poissonverteilte Zufallsvariable.
DPROBN(IRCUM,IRVAL,IS)	Diese Funktion liefert eine Zufallszahl, die aus einer diskreten Wahrscheinlichkeitsverteilung auf Basis des Zufallszahlenstroms „IS“ generiert wird, wobei das ARRAY „IRCUM“ die kumulierten Wahrscheinlichkeiten und das ARRAY „IRVAL“ die zugehörigen Zufallswerte enthält.

Tab. A-2.3.4.2: Liste sämtlicher Zufallsgenerator-Funktionen in Visual SLAM²¹⁶

²¹⁵ Zur Generierung von Pseudozufallszahlen stehen in Visual SLAM 100 verschiedene Zufallszahlenströme zur Verfügung; vgl. Pritsker 1997 /Reference/ S. 70.

²¹⁶ Weitestgehend entnommen aus Pütz 2004 /Produktionssysteme/ S. 602 in Anlehnung an Pritsker Corporation 1997 /Reference/ S. 9, Table 2.6.

(3) Liste sämtlicher intrinsischer Funktionen in Visual SLAM

Funktion	Bedeutung
ABS(RVAL)	Diese Funktion liefert den Absolutbetrag des Wertes „RVAL“.
ACOS(RVAL)	Diese Funktion liefert den Arcuskosinus des Wertes „RVAL“.
ASIN(RVAL)	Diese Funktion liefert den Arcussinus des Wertes „RVAL“.
ATAN(RVAL)	Diese Funktion liefert den Arcustangens des Wertes „RVAL“.
ATAN2(RVAL)	Diese Funktion liefert den Arcustangens des Quotienten „A/B“.
ATOI(SZVAL)	Diese Funktion konvertiert die Zeichenkette „SZVAL“ in einen ganzzahligen Wert.
ATOF(SZVAL)	Diese Funktion konvertiert die Zeichenkette „SZVAL“ in einen doppelt genauen Gleitpunktwert.
CEIL(RVAL)	Diese Funktion liefert den kleinsten ganzzahligen Wert $\geq$ „RVAL“ als doppelt genauen Gleitpunktwert.
CHAR(IVAL)	Diese Funktion konvertiert den ganzzahligen Wert „IVAL“ in ASCII Zeichen.
COS(RADIAN)	Diese Funktion liefert den Kosinus des Winkels „RADIAN“.
COSH(RADIAN)	Diese Funktion liefert den Kosinus Hyperbolicus des Winkels „RADIAN“.
DBLE(IVAL)	Diese Funktion konvertiert den ganzzahligen Wert „IVAL“ in einen doppelt genauen Gleitpunktwert.
DIM(RV1,RV2)	Diese Funktion liefert die positive Differenz zwischen „RV1“ und „RV2“.
EXP(RVAL)	Diese Funktion liefert den Wert der Exponentialfunktion e^x für „ $x = \text{RVAL}$ “.
FLOOR(RVAL)	Diese Funktion liefert den größten ganzzahligen Wert $\leq$ „RVAL“ als doppelt genauen Gleitpunktwert.
FMOD(RV1,RV2)	Diese Funktion liefert den Rest der Division „RV1/RV2“.
FTOA(RVAL)	Diese Funktion konvertiert den doppelt genauen Gleitpunktwert „RVAL“ in eine Zeichenkette.
IILEN(SZVAL)	Diese Funktion liefert die Länge der Zeichenkette „SZVAL“, wobei trennende Leerzeichen nicht mitgezählt werden.
INDEX(SZ1,SZ2)	Diese Funktion liefert die Anfangsposition (bzw. den Offset) der Zeichenkette „SZ2“ innerhalb der Zeichenkette „SZ1“.
INT(RVAL)	Diese Funktion liefert den ganzzahligen Teil von „RVAL“; der nichtganzzahlige Teil wird abgeschnitten.
ITOA(IVAL)	Diese Funktion konvertiert den ganzzahligen Wert „IVAL“ in eine Zeichenkette.
LDEXP(RVAL,IEXP)	Diese Funktion berechnet den Ausdruck „ $RVAL \cdot 2^{IEXP}$ “ und liefert das dazugehörige Ergebnis.
LEN(SZVAL)	Diese Funktion liefert die Länge der Zeichenkette „SZVAL“.
LLE(SZ1,SZ2)	Diese Funktion liefert den Wert WAHR, falls die Zeichenkette „SZ1“ $\leq$ „SZ2“.
LGE(SZ1,SZ2)	Diese Funktion liefert den Wert WAHR, falls die Zeichenkette „SZ1“ $\geq$ „SZ2“.
LGT(SZ1,SZ2)	Diese Funktion liefert den Wert WAHR, falls die Zeichenkette „SZ1“ $>$ „SZ2“.
LLT(SZ1,SZ2)	Diese Funktion liefert den Wert WAHR, falls die Zeichenkette „SZ1“ $<$ „SZ2“.

Funktion	Bedeutung
LOG(RVAL)	Diese Funktion liefert den natürlichen Logarithmus der Zahl „RVAL“.
LOG10(RVAL)	Diese Funktion liefert den Logarithmus zur Basis 10 der Zahl „RVAL“.
MAX(VAL1,VAL2)	Diese Funktion liefert das Maximum der Werte „VAL1“ und „VAL2“.
MIN(VAL1,VAL2)	Diese Funktion liefert das Minimum der Werte „VAL1“ und „VAL2“.
MOD(IVAL,IBASE)	Diese Funktion liefert den Rest der Division „IVAL/IBASE“.
NINT(RVAL)	Diese Funktion liefert den gerundeten Wert von „RVAL“; gerundet wird zum nächsten ganzzahligen Wert.
POW(RV1,RV2)	Diese Funktion liefert das Ergebnis des Ausdrucks „RV1 ^{RV2} “.
SIGN(RV1,RV2)	Diese Funktion liefert den Absolutbetrag des Wertes „RV1“ mit dem Vorzeichen des Wertes „RV2“.
SIN(RADIAN)	Diese Funktion liefert den Sinus des Winkels „RADIAN“.
SINH(RADIAN)	Diese Funktion liefert den Sinus Hyperbolicus des Winkels „RADIAN“.
SQRT(RVAL)	Diese Funktion liefert die Quadratwurzel der Zahl „RVAL“.
STRCAT(SZ1,SZ2)	Diese Funktion liefert als zusammengesetzte Zeichenkette die Zeichenkette „SZ2“, angeknüpft an die Zeichenkette „SZ1“.
TAN(RADIAN)	Diese Funktion liefert den Tangens des Winkels „RADIAN“.
TANH(RADIAN)	Diese Funktion liefert den Tangens Hyperbolicus des Winkels „RADIAN“.

Tab. A-2.3.4.3: Liste sämtlicher intrinsischer Funktionen in Visual SLAM²¹⁷

(4) Statistische Funktionen in Visual SLAM

In Visual SLAM werden während eines Simulationslaufs diverse statistische Werte ermittelt. Auf diese Werte kann mit Hilfe der statistischen Funktionen in Visual SLAM benutzerspezifisch – etwa zur Überprüfung von Bedingungen im Kontext von Entscheidungsprozessen – zugegriffen werden²¹⁸. Die ersten beiden Zeichen des Funktionsnamens dieser Funktionskategorie kennzeichnen jeweils einen von sieben verschiedenen statistischen Auswertungsbereichen gewidmeten Variablentyp in Visual SLAM; diese sind im Einzelnen²¹⁹:

- **auf Beobachtungen basierende Variablen** (Funktionsname beginnend mit „CC“): die betreffenden Variablen werden jeweils in einem COLCT-Knoten mit dem Parameter VALUE spezifiziert;

²¹⁷ Weitestgehend entnommen aus Pütz 2004 /Produktionssysteme/ S. 603 f. in Anlehnung an Pritsker Corporation 1997 /Reference/ S. 10, Table 2.7 und vom Autor ins Deutsche übersetzt.

²¹⁸ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 94 u. S. 703 ff.

²¹⁹ Weitgehend entnommen aus Pütz 2004 /Produktionssysteme/ S. 375 in Anlehnung an Pritsker, O'Reilly 1999 /Visual SLAM/ S. 703.

- **permanent gemessene Variablen** (Funktionsname beginnend mit „TT“): die betreffenden Variablen werden mit Hilfe des TIMST-Befehls spezifiziert;
- **Warteschlangendateien betreffende Variablen** (Funktionsname beginnend mit „FF“);
- **Ressourcen-spezifische Variablen** (Funktionsname beginnend mit „RR“);
- **Ressourcengruppen-spezifische Variablen** (Funktionsname beginnend mit „GR“);
- **Aktivitäten-spezifische Variablen** (Funktionsname beginnend mit „AA“);
- **Sperren-spezifische Variablen** (Funktionsname beginnend mit „GG“).

In der nachfolgenden Tabelle sind sämtliche in Visual SLAM verfügbaren statistischen Funktionen aufgelistet.

Funktion	Bedeutung
AAAVG(I)	Diese Funktion liefert den aktuell unter Berücksichtigung der letzten Löschung der Statistik ermittelten arithmetischen Mittelwert der Nutzung bzw. Auslastung der Aktivität mit der Nummer „I“.
AAMAX(I)	Diese Funktion liefert den aktuell unter Berücksichtigung der letzten Löschung der Statistik ermittelten maximalen Auslastungswert der Aktivität mit der Nummer „I“.
AASTD(I)	Diese Funktion liefert den aktuell unter Berücksichtigung der letzten Löschung der Statistik ermittelten arithmetischen Mittelwert der Nutzung bzw. Auslastung der Aktivität mit der Nummer „I“.
AATLC(I)	Diese Funktion liefert den letzten Zeitpunkt der Änderung der Auslastung der Aktivität mit der Nummer „I“.
CCAVG(I)	Diese Funktion liefert den arithmetischen Mittelwert der im Parameter „VALUE“ des COLCT-Knotens mit der Nummer „I“ seit der letzten Löschung der statistischen Auswertung erfassten Daten.
CCMAX(I)	Diese Funktion liefert den Maximalwert der im Parameter „VALUE“ des COLCT-Knotens mit der Nummer „I“ seit der letzten Löschung der statistischen Auswertung erfassten Daten.
CCMIN(I)	Diese Funktion liefert den Minimalwert der im Parameter „VALUE“ des COLCT-Knotens mit der Nummer „I“ seit der letzten Löschung der statistischen Auswertung erfassten Daten.
CCNUM(I)	Diese Funktion liefert die Anzahl der im Parameter „VALUE“ des COLCT-Knotens mit der Nummer „I“ seit der letzten Löschung der statistischen Auswertung erfassten Daten.
CCSTD(I)	Diese Funktion liefert die Standardabweichung der im Parameter „VALUE“ im COLCT-Knoten mit der Nummer „I“ seit der letzten Löschung der statistischen Auswertung erfassten Daten.
FFAVG(I)	Diese Funktion liefert die durchschnittliche, seit der letzten Löschung der statistischen Auswertung erfasste Warteschlangenlänge der Warteschlangendatei „I“.

Funktion	Bedeutung
FFAWT(I)	Diese Funktion liefert die durchschnittliche, seit der letzten Löschung der statistischen Auswertung erfasste Wartezeit der Einheiten in der Warteschlangendatei „I“.
FFMAX(I)	Diese Funktion liefert die maximale, seit der letzten Löschung der statistischen Auswertung erfasste Warteschlangenlänge der Warteschlangendatei „I“.
FFPRD(I)	Diese Funktion liefert die Zeitdauer seit der letzten Löschung der statistischen Auswertung der Warteschlangendatei I.
FFSTD(I)	Diese Funktion liefert die Standardabweichung der Warteschlangenlänge der Warteschlange „I“ seit der letzten Löschung der statistischen Auswertung.
FFTLC(I)	Diese Funktion liefert den letzten Zeitpunkt der Änderung der Warteschlangenlänge der Warteschlangendatei „I“.
GGOPN(I)	Diese Funktion liefert den prozentualen Anteil des Status „OPEN“ der Sperre „I“ an der Zeitdauer seit der letzten Löschung der statistischen Auswertung.
GGTLC(I)	Diese Funktion liefert den Zeitpunkt der letzten Änderung des Status der Sperre „I“ seit der letzten Löschung der statistischen Auswertung.
GRPAVA(I)	Diese Funktion liefert die durchschnittliche Verfügbarkeit der Ressourcengruppe „I“ seit der letzten Löschung der statistischen Auswertung.
GRPAVG(I)	Diese Funktion liefert die durchschnittliche Auslastung der Ressourcengruppe „I“ seit der letzten Löschung der statistischen Auswertung.
GRPMAX(I)	Diese Funktion liefert die maximale Auslastung der Ressourcengruppe „I“ seit der letzten Löschung der statistischen Auswertung.
GRPPRD(I)	Diese Funktion liefert die Zeitdauer seit der letzten Löschung der statistischen Auswertung der Ressourcengruppe „I“.
GRPSTD(I)	Diese Funktion liefert die Standardabweichung der Auslastung der Ressourcengruppe „I“ seit der letzten Löschung der statistischen Auswertung.
GRPTLC(I)	Diese Funktion liefert den Zeitpunkt der letzten Veränderung der Auslastung der Ressourcengruppe „I“ seit der letzten Löschung der statistischen Auswertung.
RRAVA(I)	Diese Funktion liefert die durchschnittliche Verfügbarkeit der Ressource „I“ seit der letzten Löschung der statistischen Auswertung.
RRAVG(I)	Diese Funktion liefert die durchschnittliche Auslastung der Ressource „I“ seit der letzten Löschung der statistischen Auswertung.
RRMAX(I)	Diese Funktion liefert die maximale Auslastung der Ressource „I“ seit der letzten Löschung der statistischen Auswertung.
RRPRD(I)	Diese Funktion liefert die Zeitdauer seit der letzten Löschung der statistischen Auswertung der Ressource „I“.
RRSTD(I)	Diese Funktion liefert die Standardabweichung der Auslastung der Ressource „I“ seit der letzten Löschung der statistischen Auswertung.

Funktion	Bedeutung
RRTLC(I)	Diese Funktion liefert den Zeitpunkt der letzten Veränderung der Auslastung der Ressource „I“ seit der letzten Löschung der statistischen Auswertung.
TTAVG(I)	Diese Funktion liefert den seit der letzten Löschung der statistischen Auswertung ermittelten durchschnittlichen Wert der permanent gemessenen Variablen, deren „TIMST“-Spezifikation mit dem Statistikindex „I“ gekennzeichnet ist.
TTMAX(I)	Diese Funktion liefert den seit der letzten Löschung der statistischen Auswertung ermittelten maximalen Wert der permanent gemessenen Variablen, deren „TIMST“-Spezifikation mit dem Statistikindex „I“ gekennzeichnet ist.
TTMIN(I)	Diese Funktion liefert den seit der letzten Löschung der statistischen Auswertung ermittelten minimalen Wert der permanent gemessenen Variablen, deren „TIMST“-Spezifikation mit dem Statistikindex „I“ gekennzeichnet ist.
TTPRD(I)	Diese Funktion liefert die Zeitdauer seit der letzten Löschung der statistischen Auswertung der permanent gemessenen Variablen, deren „TIMST“-Spezifikation mit dem Statistikindex „I“ gekennzeichnet ist.
TTSTD(I)	Diese Funktion liefert die seit der letzten Löschung der statistischen Auswertung ermittelte Standardabweichung des Wertes der permanent gemessenen Variablen, deren „TIMST“-Spezifikation mit dem Statistikindex „I“ gekennzeichnet ist.
TTTLC(I)	Diese Funktion liefert den Zeitpunkt der letzten Veränderung der permanent gemessenen Variablen, deren „TIMST“-Spezifikation mit dem Statistikindex „I“ gekennzeichnet ist.

Tab. A-2.3.4.4: Liste sämtlicher statistischer Funktionen in Visual SLAM²²⁰

A-2.4 Zu grundlegenden subnetzwerkbezogenen Sprachelementen in Visual SLAM

A-2.4.1 Variablen

Wie in Abschnitt 3.2.2.2.1.2 unter (4) erwähnt, können in OSN grundsätzlich sämtliche in Visual SLAM-Netzwerken verfügbare Variablen genutzt werden. Nachfolgend werden die den OSN-Instanzen vorbehaltenen lokalen einheitenspezifischen sowie lokalen benutzerspezifisierbaren Variablen skizziert.

Bei den lokalen einheitenspezifischen Variablen handelt es sich um Attribute der (lokalen) Einheiten einer OSN-Instanz²²¹, weshalb diese auch als lokale Attribute bezeichnet werden²²². Die lokalen benutzerspezifisierbaren Variablen entsprechen weitgehend den oben in A-2.2

²²⁰ Weitgehend entnommen aus Pütz 2004 /Produktionssysteme/ S. 605 ff. in Anlehnung an Pritsker Corporation 1997 /Reference/ S. 11, Table 2.8.

²²¹ Vgl. hierzu Pritsker Corporation 1997 /Reference/ S. 6, Table 2.3; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 287.

²²² Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 280.

unter (2) skizzierten globalen benutzerspezifisierbaren Variablen in Visual SLAM Netzwerken, ihr Geltungsbereich ist jedoch auf die zugehörige (lokale) OSN-Instanz beschränkt²²³. Die einzelnen OSN-spezifischen lokalen Variablenarten werden in der nachfolgenden Tabelle angeführt.

Variable	Definition
XXINST[I]	reelle lokale benutzerspezifizierte Variable „I“ einer OSN-Instanz; die maximale Variablenanzahl „I“ dieser Variablenart ist in der „LIMITSVSN“-Anweisung ²²⁴ des zugehörigen OSN zu spezifizieren
LLINST[I]	ganzzahlige benutzerspezifizierte Variable I einer OSN-Instanz; die maximale Variablenanzahl „I“ dieser Variablenart ist in der „LIMITSVSN“-Anweisung des zugehörigen OSN zu spezifizieren
SZINST[I]	globale benutzerspezifizierte Zeichenketten-Variable „I“ einer OSN-Instanz; die maximale Variablenanzahl „I“ dieser Variablenart ist in der „LIMITSVSN“-Anweisung des zugehörigen OSN zu spezifizieren
ANTRIB[I]	reelles Attribut „I“ der aktuellen lokalen Einheit einer OSN-Instanz; die maximale Attributeanzahl „I“ dieser Variablenart ist in der „LIMITSVSN“-Anweisung des zugehörigen OSN zu spezifizieren
LNTRIB[I]	ganzzahliges Attribut I der aktuellen lokalen Einheit einer OSN-Instanz; die maximale Attributeanzahl „I“ dieser Variablenart ist in der „LIMITSVSN“-Anweisung des zugehörigen OSN zu spezifizieren
SNTRIB[I]	Zeichenketten-Attribut „I“ der aktuellen lokalen Einheit einer OSN-Instanz; die maximale Attributeanzahl I dieser Variablenart ist in der LIMITSVSN-Anweisung des zugehörigen OSN zu spezifizieren

Tab. A-2.4.1: OSN-spezifische lokale globale Variablenarten und lokale Attribute in Visual SLAM²²⁵

A-2.4.2 Netzwerk- und Subnetzwerkelemente

Die Visual SLAM Netzwerkelemente stehen grundsätzlich auch zur Modellierung in OSN zur Verfügung; ihre dortige Nutzung unterliegt jedoch folgenden Restriktionen²²⁶:

- CREATE-Knoten und Enter-Knoten²²⁷ sind in OSN nicht zulässig.
- Die mit Hilfe eines DETECT-Knotens in einem OSN erzeugten Einheiten dürfen nicht zu einem RETURNVSN-Knoten²²⁸ geleitet werden.

²²³ Vgl. hierzu Pritsker Corporation 1997 /Reference/ S. 6, Table 2.2; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 287.

²²⁴ Die „LIMITSVSN“-Anweisung wird in Abschnitt Anhang A-2.6.2 erläutert.

²²⁵ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 378 in Anlehnung an Pritsker, O'Reilly 1999 /Visual SLAM/ S. 287; Pritsker Corporation 1997 /Reference/ S. 6, Table 2.2 u. Table 2.3.

²²⁶ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 287 f.

²²⁷ Der ENTER-Knoten wird im Anhang A-2 in Abschnitt A-2.5.1 in (2) erläutert.

²²⁸ Der RETURNVSN-Knoten wird im aktuellen Abschnitt in (4) erläutert.

- Bei einem ACCUMULATE-Knoten²²⁹ innerhalb eines OSN dient der Parameter SAVE dazu, den zugehörigen Einheiten-Kontext zu spezifizieren. Kontext einer Einheit können dabei entweder ihre Hauptnetzwerk-Attribute oder ihre lokalen Attribute der jeweiligen OSN-Instanz sein²³⁰.

Die zur OSN-Instanzierung zulässigen Netzwerkelemente sind in der nachfolgenden Tabelle nebst Angabe ihrer zur Instanzierung zulässigen Parameter aufgelistet.

Netzwerkelement	zulässige Parameter für OSN-Instanzierung
RESOURCE-Block	CAP (Ressourcenkapazität)
SERVICE-/ NONSERVICE-Aktivität	Duration (Verzögerungsdauer) PROB (Wahrscheinlichkeit) COND (Bedingung)
ALTER-Knoten	CC (Veränderung der Ressourcenkapazität)
ASSIGN-Knoten	VAR (Variable) VALUE (VAR zuzuweisender Wert)
AWAIT-Knoten	IFL (Warteschlangendatei) UR [Anzahl anzufordernder (Ressourcen-) Kapazitätseinheiten]
FREE-Knoten	UF [Anzahl freizusetzender (Ressourcen-) Kapazitätseinheiten]
PREEMPT-Knoten	IFL (Warteschlangendatei) PR (Priorisierungsattribut) NATR (Attribut zur Speicherung der Restbedienungszeit)
QUEUE-Knoten	IFL (Warteschlangendatei)
TERMINATE-Knoten	TC [Anzahl zu zerstörender Einheiten (Terminierungszähler)]

Tab. A-2.4.2.1: Zur OSN-Instanzierung zulässige Netzwerkelemente mit ihren zur OSN-Instanzierung zulässigen Parametern in Visual SLAM²³¹

Im Folgenden werden diejenigen (Sub-) Netzwerkelemente erläutert, welche zur Definition eines OSN (VSN-Block), zum Aufruf einer OSN-Instanz (CALLVSN-Knoten), zum Einbringen einer Einheit in eine OSN-Instanz (ENTERVSN-Knoten) und zum Zurückleiten einer Einheit aus einer OSN-Instanz (RETURNVSN-Knoten) dienen²³².

²²⁹ Hinsichtlich des ACCUMULATE-Knotens sei verwiesen auf Pritsker, O'Reilly 1999 /Visual SLAM/ S. 240 ff.; Pritsker Corporation 1997 /Reference/ S. 12 f.

²³⁰ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 287.

²³¹ Entnommen aus Pütz 2004 /Produktionssysteme/ S. in Anlehnung an Pritsker, O'Reilly 1999 /Visual SLAM/ S. 281, Table 8.1.

²³² Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 285 ff.

(1) VSN-Block

<u>Subnetzwerksymbol:</u>		<u>Programmanweisung in Visual SLAM:</u>			
<table border="1"> <tr> <td>VSNNAME</td><td>PNAME, PTYPE, PDESC</td><td>repeats ...</td></tr> </table>		VSNNAME	PNAME, PTYPE, PDESC	repeats ...	VSN,VSNNAME,{ {PNAME,PTYPE,PDESC} ,repeats};
VSNNAME	PNAME, PTYPE, PDESC	repeats ...			
Eingabefeld	Option	Voreinstellung			
VSNNAME	Zeichenkette	Angabe erforderlich			
PNAME	Zeichenkette	unbesetzt			
PTYPE	DOUBELVAL, DOUBLEREF, LONGVAL, LONGREF, STRINGVAL oder STRINGREF	DOUBELVAL			
PDESC	Zeichenkette	unbesetzt			

Abb. A-2.4.2.1: VSN-Block mit Parameterliste und Erläuterungen²³³

Der VSN-Block dient jeweils zur Spezifikation des OSN-Namens „VSNNAME“ sowie den in der entsprechenden OSN-Klasse zur OSN-Instanzierung zu verwendenden (lokalen) Parametern²³⁴. Dabei ist für jeden dieser Parameter mit „PTYPE“ der Datentyp zu spezifizieren. In der nachfolgenden Tabelle sind die Optionen zur Spezifikation von „PTYPE“ aufgelistet.

PTYPE-OPTION	Definition
DOUBLEVAL	Doppeltgenauer Gleitpunktwert, der <u>nicht</u> innerhalb des das OSN aufrufenden (Sub-) Netzwerkteils verändert werden darf.
DOUBELREF	Doppeltgenauer Gleitpunktwert, der innerhalb des das OSN aufrufenden (Sub-) Netzwerkteils verändert werden darf.
LONGVAL	Langer ganzzahliger Wert, der <u>nicht</u> innerhalb des das OSN aufrufenden (Sub-) Netzwerkteils verändert werden darf.
LONGREF	Langer ganzzahliger Wert, der innerhalb des das OSN aufrufenden (Sub-) Netzwerkteils verändert werden darf.
STRINGVAL	Zeichenkette, die <u>nicht</u> innerhalb des das OSN aufrufenden (Sub-) Netzwerkteils verändert werden darf.
STRINGREF	Zeichenkette, die innerhalb des das OSN aufrufenden (Sub-) Netzwerkteils verändert werden darf.

Tab. A-2.4.2.2: Optionen zur Spezifikation von PTYPE innerhalb des VSN-Blocks in Visual SLAM²³⁵

²³³ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 380; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 282 f.; Pritsker Corporation 1997 /Reference/ S. 52 f.

²³⁴ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 282 f.; Pritsker Corporation 1997 /Reference/ S. 52 f.

²³⁵ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 282 f.; Pritsker Corporation 1997 /Reference/ S. 52.

Die (Instanzierungs-) Parameter können durch Spezifikation von „PNAME“ bezeichnet werden. Wird stattdessen die Voreinstellung von „PNAME“ beibehalten, so wird den betreffenden Parametern in Visual SLAM mit „PP[I]“ die Vektorposition „I“ im Vektor „PP“ zugewiesen, wobei „I“ der Listenposition des jeweiligen Parameters gemäß der zugrunde liegenden Eingabereihenfolge entspricht. Zur Erläuterung der Bedeutung der Parameter kann mit „PDESC“ eine benutzerspezifisierbare Zeichenkette verwendet werden²³⁶.

(2) CALLVSN-Knoten

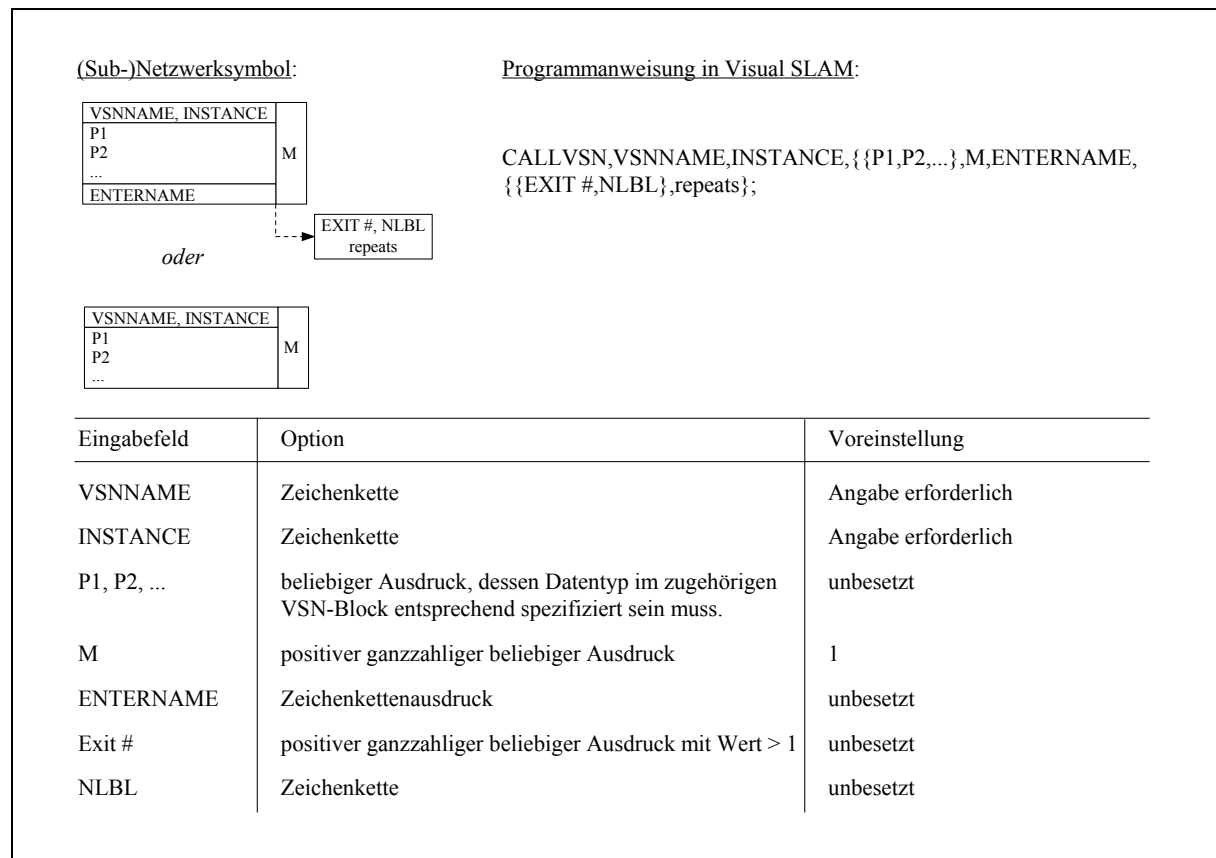


Abb. A-2.4.2.2: CALLVSN-Knoten mit Parameterliste und Erläuterungen²³⁷

Der CALLVSN-Knoten dient jeweils zum Aufruf einer OSN-Instanz; dabei wird der jeweilige Aufruf durch das Eintreffen einer Einheit am betreffenden CALLVSN-Knoten ausgelöst²³⁸. Dazu ist im CALLVSN-Knoten mit „VSNNNAME“ zunächst die OSN-Bezeichnung zu spezifizieren. Mit „INSTANCE“ kann die Instanzbezeichnung der aufzurufenden OSN-Instanz vereinbart werden²³⁹. Wird „INSTANCE“ dagegen nicht spezifiziert, so wird beim

²³⁶ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 282; Pritsker Corporation 1997 /Reference/ S. 52.

²³⁷ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 381; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 288 f.; Pritsker Corporation 1997 /Reference/ S. 22 f.

²³⁸ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 280 f. u. S. 288.

²³⁹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 288.

Eintreffen einer Einheit am entsprechenden CALLVSN-Knoten die „unbezeichnete“ Instanz der zugrunde liegenden OSN-Klasse aufgerufen. Beim Aufruf einer OSN-Instanz kann durch Spezifikation von „P1“, „P2“ usw. eine Liste von Parameterwerten in die jeweils aufgerufene OSN-Instanz übertragen werden²⁴⁰. Dazu müssen die Datentypen der zu übertragenden Parameterwerte mit denjenigen Datentypen übereinstimmen, die im zugehörigen VSN-Block (mit Hilfe von „PTYPE“) vereinbart wurden²⁴¹.

Darüber hinaus kann mit dem Parameter „ENTERNAME“ ein bestimmter Eingangsknoten (ENTERVSN-Knoten²⁴²) der aufgerufenen OSN-Instanz gewählt werden, zu dem die jeweils am zugrunde liegenden CALLVSN-Knoten eintreffende Einheit entsprechend weitergeleitet werden soll. Wird demgegenüber „ENTERNAME“ nicht spezifiziert, erfolgt die Weiterleitung der betreffenden Einheiten zum jeweils ersten der im zugehörigen OSN modellierten ENTERVSN-Knoten²⁴³.

Einheiten, die von einem OSN zurück zum aufrufenden (Sub-) Netzwerkteil geleitet werden, kehren dort zu demjenigen CALLVSN-Knoten zurück, von dem aus sie ursprünglich zum OSN geleitet wurden. Dort eingetroffen, werden sie unmittelbar im zugehörigen (Sub-) Netzwerkteil weitergeleitet. Hierzu können unterschiedliche Ausgänge des CALLVSN-Knotens unter Angabe der Ausgangsnummer im Parameter „Exit #“ mit zugehörigen Spezifikationen vom Zielknoten „NLBL“ im aktuellen (Sub-) Netzwerkteil vereinbart werden²⁴⁴. Wird auf die Spezifikation solcher alternativer Ausgänge verzichtet, erfolgt die Weiterleitung der Einheiten unter Berücksichtigung der vom CALLVSN-Knoten ausströmenden Aktivitäten und der Spezifikation seines M-Parameters²⁴⁵.

CALLVSN-Knoten können sowohl Bestandteil von Netzwerken als auch von OSN sein, so dass letztere jeweils auch aus einem übergeordneten OSN heraus aufgerufen werden können.

²⁴⁰ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 288 f.

²⁴¹ Vgl. Pritsker Corporation 1997 /Reference/ S. 23.

²⁴² Vgl. hierzu Abschnitt 4.2.1.4.3.3.

²⁴³ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 288.

²⁴⁴ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 288. Siehe dazu auch die obere Symbolalternative des CALLVSN-Knotens in Abb. A-2.4.2.2.

²⁴⁵ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 288. Siehe dazu auch die untere Symbolalternative des CALLVSN-Knotens in Abb. A-2.4.2.2.

(3) ENTERVSN-Knoten

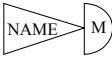
<u>Subnetzwerksymbol:</u>			<u>Programmanweisung in Visual SLAM:</u>		
			ENTERVSN,NAME,M;		
Eingabefeld	Option		Voreinstellung		
NAME	Zeichenkette		unbesetzt		
M	positiver ganzzahliger beliebiger Ausdruck		1		

Abb. A-2.4.2.3: ENTERVSN-Knoten mit Parameterliste und Erläuterungen²⁴⁶

Die ENTERVSN-Knoten dienen in OSN jeweils als Eingangsknoten und können durch Spezifizierung des Parameters „NAME“ bezeichnet werden. OSN können dabei mehrere solcher ENTERVSN-Knoten enthalten²⁴⁷.

Damit Einheiten jeweils aus einem übergeordneten (Sub-) Netzwerk zu einem ENTERVSN-Knoten geleitet werden können, ist es erforderlich, dass die Spezifikation des Parameters „ENTERNAME“ des CALLVSN-Knotens²⁴⁸ im aufrufenden (Sub-) Netzwerkteil mit der Spezifikation des Parameters „NAME“ des betreffenden ENTERVSN-Knotens übereinstimmen. Werden die genannten Parameter „ENTERNAME“ und „NAME“ nicht spezifiziert, so werden die am zugehörigen CALLVSN-Knoten eintreffenden Einheiten jeweils zum ersten im betreffenden OSN modellierten ENTERVSN-Knoten geleitet²⁴⁹.

²⁴⁶ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 383; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 285 f.; Pritsker Corporation 1997 /Reference/ S. 30.

²⁴⁷ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 285 f.; Pritsker Corporation 1997 /Reference/ S. 30.

²⁴⁸ Hinsichtlich des CALLVSN-Knotens sei verwiesen auf die im aktuellen Abschnitt zuvor unter (2) gewährten Hinweise.

²⁴⁹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 286.

(4) RETURNVSN-Knoten

Subnetzwerksymbol:

ERETURN

Exit#

Programmanweisung in Visual SLAM:

RETURNVSN,ERETURN,Exit#,M;

Eingabefeld	Option	Voreinstellung
ERETURN	beliebiger Ausdruck	0.0
Exit#	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.4.2.4: RETURNVSN-Knoten mit Parameterliste und Erläuterungen²⁵⁰

Der RETURNVSN-Knoten dient jeweils dazu, Einheiten aus einer OSN-Instanz zu jenem CALLVSN-Knoten zurückzuleiten, welcher die zugrunde liegende OSN-Instanz ursprünglich aufgerufen hat²⁵¹. Im RETURNVSN-Knoten können dazu die Parameter „ERETURN“ und „EXIT#“ spezifiziert werden. Mit dem Parameter „ERETURN“ kann der zurückzuleitenden Einheit ein Wert als spezielles Attribut (bzw. einheitenspezifische Variable) zugeordnet werden. Mit dem Parameter „Exit#“ kann die Nummer des Ausgangs des betreffenden CALLVSN-Knotens spezifiziert werden. Wird dabei die Voreinstellung von „Exit#“ gewählt oder als Wert Null spezifiziert, so erfolgt die Weiterleitung der Einheiten unter Berücksichtigung der vom CALLVSN-Knoten ausströmenden Aktivitäten und der Spezifikation seines M-Parameters²⁵².

²⁵⁰ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 383; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 286; Pritsker Corporation 1997 /Reference/ S. 46.

²⁵¹ Vgl. hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 286; Pritsker Corporation 1997 /Reference/ S. 46.

²⁵² Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 286. Vgl. hierzu auch die Erläuterungen zum CALLVSN-Knoten im aktuellen Abschnitt unter (2).

A-2.4.3 Funktionen

Die unter (3) in (c) des Abschnitts 3.2.2.2.1.2 skizzierten und in Anhang A-2.3.4 aufgelisteten Funktionen der Visual SLAM Netzwerkebene stehen grundsätzlich auch zur Verwendung in OSN zur Verfügung. Dies liegt nicht zuletzt daran, dass OSN in vergleichbarer Art und Weise wie Visual SLAM Netzwerke modelliert werden²⁵³.

Speziell für OSN steht in Visual SLAM zusätzlich die Funktion „SIZE(PARRAY)“ zur Verfügung²⁵⁴. Diese dient dazu, die Anzahl der Elemente einer Parameterliste „PARRAY“, welche beim Aufruf einer OSN-Instanz von einem CALLVSN-Knoten²⁵⁵ an diese OSN-Instanz übertragen wird, wiederzugeben.

A-2.5 Zu User Insert-spezifischen Sprachelementen in Visual SLAM in Verbindung mit der Programmiersprache C

A-2.5.1 Spezielle Netzwerk- und Subnetzwerkelemente

(1) EVENT-Knoten

<u>Subnetzwerksymbol:</u>		<u>Programmanweisung in Visual SLAM:</u>
		EVENT,JEVNT,M;
Eingabefeld	Option	Voreinstellung
JEVNT	ganzzahliger beliebiger Ausdruck	Angabe erforderlich
M	positiver ganzzahliger beliebiger Ausdruck	1

Abb. A-2.5.1.1: EVENT-Knoten mit Parameterliste und Erläuterungen²⁵⁶

Der EVENT-Knoten dient jeweils als Schnittstelle zwischen einem Visual SLAM (Sub-) Netzwerk und benutzerdefinierten Eingaben, die entweder in Form eines C- oder eines Visual Basic-Programms implementiert sind. Trifft jeweils eine Einheit an einem solchen EVENT-Knoten ein, wird im szenariospezifisch entsprechend eingebundenen C- oder Visual Basic-

²⁵³ Vgl. ähnlich hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 280.

²⁵⁴ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 287.

²⁵⁵ Hinsichtlich des CALLVSN-Knotens sei verwiesen auf die im aktuellen Abschnitt zuvor unter Punkt (2) erfolgten Hinweise.

²⁵⁶ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 385; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 229 u. S. 544 f.; Pritsker Corporation 1997 /Reference/ S. 31.

Programm die „EVENT“-Funktion²⁵⁷ aufgerufen und die im jeweiligen Programm zur Ereignis-Nummer „JEVNT“ codierten Programmanweisungen werden ausgeführt²⁵⁸. Die den Funktionsaufruf jeweils auslösende Einheit wird daraufhin unter Berücksichtigung des am zugrunde liegenden EVENT-Knoten spezifizierten Parameters „M“ und der etwaigen, von diesem Knoten ausströmenden Aktivitäten weitergeleitet.

(2) ENTER-Knoten

<u>Subnetzwerksymbol:</u>			<u>Programmanweisung in Visual SLAM:</u>		
			ENTER,NUM,M;		
Eingabefeld	Option		Voreinstellung		
NUM	positiver ganzzahliger konstanter Ausdruck		Angabe erforderlich		
M	positiver ganzzahliger beliebiger Ausdruck		1		

Abb. A-2.5.1.2: ENTER-Knoten mit Parameterliste und Erläuterungen²⁵⁹

Der ENTER-Knoten dient jeweils dazu, Einheiten aus einer benutzerspezifisierten Funktion, die in einem szenariospezifisch eingebundenen C- oder Visual Basic-Programm enthalten ist, selektiv mit Hilfe der ENTER-Funktion in ein Netzwerk einzufügen²⁶⁰. Hierzu muss die ganzzahlige Konstante „NUM“ des ENTER-Knotens in der entsprechenden ENTER-Funktion spezifiziert sein²⁶¹. Jede Einheit, die aus einem szenariospezifisch eingebundenen C- bzw. Visual Basic-Programm an einem ENTER-Knoten in ein Netzwerk eingefügt wird, wird unmittelbar unter Berücksichtigung des im ENTER-Knoten spezifizierten „M“-Parameters und der etwaigen zugehörigen ausströmenden Aktivitäten weitergeleitet²⁶².

²⁵⁷ Die „EVENT“-Funktion ist eine benutzerspezifisierbare Visual SLAM-Funktion, deren grundlegender Aufbau in Abschnitt A-2.5.3 in (1) unter (a) beschrieben wird; vgl. zu dieser Funktion als Bestandteil von C-Programmen ferner Pritsker, O'Reilly 1999 /Visual SAM/ S. 544 ff.; Pritsker Corporation 1997 /Reference/ S. 124.

²⁵⁸ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 229; Pritsker Corporation 1997 /Reference/ S. 31.

²⁵⁹ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 386; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 229 u. S. 548; Pritsker Corporation 1997 /Reference/ S. 29.

²⁶⁰ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 230 u. S. 548; Pritsker Corporation 1997 /Reference/ S. 29.

²⁶¹ Die Funktion „ENTER“ ist eine benutzeraufrufbare Visual SLAM-Funktion, deren grundlegender Aufbau in Abschnitt A-2.5.3 in (1) unter (2) näher erläutert wird; vgl. zu dieser als Bestandteil von C-Programmen ferner Pritsker, O'Reilly 1999 /Visual SAM/ S. 548; Pritsker Corporation 1997 /Reference/ S. 124.

²⁶² Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 230. Werden Einheiten innerhalb eines Netzwerkes zu einzelnen ENTER-Knoten geleitet, werden sie von dort unmittelbar (auf der Netzwerkebene) weitergeleitet.

A-2.5.2 Benutzerspezifische Eingaben in Gestalt von C-Programmen

Für die C-Schnittstelle des Simulationssystems AweSim steht zwecks umfangreichen Zugriffs auf die Visual SLAM-Datenstrukturen und -Sprachelemente die Definitionsdatei „vslam.h“ zur Verfügung²⁶³. Auf diese Visual SLAM-Bestandteile kann mithilfe spezieller Funktionen zurückgegriffen werden, auf die in Anhang A-2.5.3 näher eingegangen wird. Im Rahmen der Erstellung benutzerspezifischer Eingaben in Form von C-Programmen sind die Visual SLAM Sprachelemente "ENTRY *" und "ENTITY *" von großer Bedeutung, da sie zur Manipulation von Einheiten bzw. einheitenspezifischen Informationen bei der Nutzung mehrerer der zuvor bereits erwähnten speziellen Funktionen erforderlich sind.

„ENTRY *“ und „ENTITY *“ sind Variablentypen, welche als Zeiger²⁶⁴ auf Strukturen²⁶⁵ dienen. „ENTRY *“ zeigt dabei jeweils auf eine Struktur, die zur Verwaltung der Position einzelner Einheiten innerhalb von (Warteschlangen-) Dateien dient²⁶⁶. ENTITY * zeigt jeweils auf eine Struktur, welche zur Verwaltung einzelner Attribute (einheitenspezifische Variablen) einer Attributart relevanter Einheiten dient, auf die (aktuell) zugegriffen wird²⁶⁷. Wird z.B. in einem Visual SLAM Netzwerk bei einer Einheit, die aktuell im Zugriff ist, mit ATTRIB[2] auf deren reelles Attribut referenziert, welches in der Vektorposition „zwei“ des „ATTRIB“-Vektors gespeichert ist, kann dies in einem C-Programm beispielsweise mithilfe der Anweisung "peUser->ATTRIB[2]" mit „peUser“ als ENTITY *-Zeiger realisiert werden²⁶⁸.

Eine Liste sämtlicher einheitenspezifischer Variablenarten, auf die mit „ENTITY *“ in C-Programmen referenziert werden kann, inklusive der zugehörigen (Variablen-) Typvereinbarung (gem. „vslam.h“), ist in der nachfolgenden Tabelle enthalten.

²⁶³ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 543 f. Diese Definitionsdatei (bzw. header file) enthält Deklarationen der Visual SLAM-Variablen und Funktionen und ist mithilfe der „#include“-Anweisung (der Programmiersprache C) in das jeweils in einem Visual SLAM Szenario einzubindende C-Programm einzufügen; vgl. zur Nutzung entsprechender Definitionsdateien in C-Programmen und der dazu erforderlichen „#include“-Anweisung z.B. Kernighan, Ritchie 1990 /C/ S. 32 u. S. 80 ff.

²⁶⁴ Unter einem Zeiger, auch pointer genannt, wird allgemein eine Variable verstanden, welche jeweils die Adresse einer Variablen (als Variablenwert) beinhaltet; vgl. hierzu z.B. Kernighan, Ritchie 1990 /C/ S. 5.

²⁶⁵ Unter einer Struktur (structure) wird gemeinhin eine Gruppe von Variablen verstanden, welche logisch zusammengehören und unter einem einzigen bzw. gemeinsamen Namen zusammengefasst werden, um als Einheit separat behandelt werden zu können; vgl. dazu Kernighan, Ritchie 1990 /C/ S. 123.

²⁶⁶ Vgl. Pritsker Corporation 1997 /Reference/ S. 119; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 543 f.

²⁶⁷ Vgl. Pritsker Corporation 1997 /Reference/ S. 119; Pritsker, O'Reilly 1999 /Visual SLAM/ S. 543 f.

²⁶⁸ Vgl. Pritsker Corporation 1997 /Reference/ S. 119.

(Variablen-)Typvereinbarung		Bedeutung
int	ETYPE	Ganzzahliges Attribut zur Vergabe einer Typennummer für die betreffende Einheit.
UINT	ENUM	Eindeutiger ganzzahliger nur-Lese-Bezeichner der betreffenden Einheit.
double *	ATTRIB	Reelle Attributwerte (vereinbart als doppelt genaue Gleitpunktwerte)
long *	LTRIB	Lange (d.h. mit 32 Bit darstellbare) ganzzahlige Attributwerte.
char **	STRIB	Zeichenkettenattributwerte.
double *	ANTRIB	Reelle Attributwerte einer lokalen Einheit einer objektorientierten Subnetzwerk (OSN)-Instanz (vereinbart als doppelt genauer Gleitpunktwert).
long *	LNTRIB	Lange (d.h. mit 32 Bit darstellbare) ganzzahlige Attributwerte einer lokalen Einheit einer OSN-Instanz.
char **	SNTRIB	Zeichenkettenattributwerte einer lokalen Einheit einer OSN-Instanz.
double **	ERETURN	Reelle einheitenspezifische Variable, deren Wert mit Hilfe des Parameters „ERETURN“ eines RETURNVSN-Knotens spezifiziert werden kann.
double	NETIME	Ereigniszeit-Attribut oder Restzeit-Attribut.

Tab. A-2.5.2: Liste sämtlicher laut (Variablen-) Typvereinbarung in der Quelldatei „vslam.h“ in einem C-Programm referenzierbaren einheitenspezifischen Variablen²⁶⁹

A-2.5.3 Spezielle Funktionen

In Visual SLAM stehen zwei Arten von speziellen Funktionen zur Verwendung in C-Programmen zur Verfügung, und zwar benutzerspezifisierbare C-Funktionen und benutzeraufrufbare C-Funktionen²⁷⁰.

(1) Ausgewählte benutzerspezifisierbare C-Funktionen

Die in Visual SLAM verfügbaren benutzerspezifisierbaren C-Funktionen²⁷¹ sind allesamt jeweils als Prototypen bzw. Dummy Versionen in der Visual SLAM Definitionsdatei „vslam.h“ enthalten²⁷². Die möglichen Resultattypen der einzelnen benutzerspezifisierbaren C-Funktionen sind in der nachfolgenden Tabelle (A-2.5.3) aufgelistet.

²⁶⁹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 545 f.; Pritsker Corporation 1997 /Reference/ S. 119.

²⁷⁰ Vgl. Pritsker Corporation 1997 /Reference/ S. 119 ff. u. S. 139 ff.

²⁷¹ Hinsichtlich einer Erläuterung sämtlicher in Visual SLAM zur Verfügung stehender benutzerspezifisierbarer C-Funktionen wird hier auf Pritsker Corporation 1997 /Reference/ S. 119 ff. verwiesen.

²⁷² Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 544.

Resultattyp	Bedeutung
void	Funktion liefert keinen Resultatwert.
double	Funktion liefert einen doppelt genauen Gleitpunktwert als Resultatwert.
UINT	Funktion liefert einen vorzeichenlosen ganzzahligen Wert als Resultatwert.
BOOL	Funktion liefert als Resultatwert einen ganzzahligen Wahrheitswert, wobei 1 wahr und 2 falsch repräsentiert.
ENTRY *	Zeiger auf eine Struktur, in welcher die Position einer Einheit innerhalb einer (Warteschlangen-) Datei verwaltet wird.
ENTITY *	Zeiger auf eine Struktur, in welcher jeweils die einzelnen Attribute einer Attributart einer Einheit verwaltet werden.

Tab. A-2.5.3: Resultattypen der in Visual SLAM zur Verfügung stehenden benutzerspezifisierbaren C-Funktionen²⁷³

Die Funktionsdefinitionen der einzelnen benutzerspezifisierbaren C-Funktionen sind durch folgenden Aufbau gekennzeichnet:

```
Resultattyp SWFUNC Funktionsname( Typ Parameter_1, Typ Parameter_2, ...)
{
    Vereinbarungen
    Anweisungen
}
```

Dabei ist „SWFUNC“ ein Schlüsselwort, welches in einem C-Compiler²⁷⁴ zur Kennzeichnung von benutzerspezifisierten Funktionen in Windows erforderlich ist²⁷⁵. Die Parameterbezeichnungen (wie z.B. „Parameter_1“ in der zuvor angeführten formalisierten Funktionsdefinition), die in den Prototypen der betreffenden benutzerspezifisierbaren C-Funktionen enthalten sind, können benutzerspezifisch verändert werden. In den nachfolgend ausgewählten Beispielen werden die betreffenden benutzerspezifisierbaren C-Funktionen jeweils in einer ihre prototypische (bzw. Dummy) Spezifikation in Visual SLAM berücksichtigenden allgemeinen Form unter Verwendung des Quellcodes der Programmiersprache C wiedergegeben²⁷⁶. Des

²⁷³ Vgl. Pritsker Corporation 1997 /Reference/ S. 119.

²⁷⁴ Für die Compilierung der benutzerspezifischen Eingaben in Form von C-Programmen wurde in dieser Arbeit Microsoft Visual C++ in der Version 5.0 verwendet.

²⁷⁵ Mit Hilfe des Schlüsselworts SWFUNC werden die benutzerspezifisierten C-Funktionen in Windows in eine dynamisch bindende Bibliothek (dynamic link library) eingefügt, was insbesondere die Auswahl von Simulationsalternativen erleichtert; vgl. hierzu Pritsker Corporation 1997 /Reference/ S. 119.

²⁷⁶ Dabei werden Kommentarzeilen den Vorgaben der Programmiersprache C entsprechend mit „/*“ eingeleitet und mit „*/“ abgeschlossen.

Weiteren werden die zulässigen Argumentwerte der dort auftretenden prototypischen Parameterbezeichnungen erläutert²⁷⁷.

(a) „EVENT“-Funktion

Allgemeine Form der Funktion EVENT:

```
void SWFUNC EVENT(int iCode, ENTITY * peUser)
{
    /* Vereinbarungen */
    switch (iCode)
    {
        case 1:      /* falls eine Einheit an dem EVENT-Knoten mit JEVNT = iCode = 1 eintrifft */
                     /* oder ein mit der Funktion SCHDL geplantes Ereignis vom Typ iEvent = iCode = 1 */
                     /* als aktuelles Ereignis laut Eintrag im Ereigniskalender von Visual SLAM eintritt */
                     /* Anweisungen */
                     break;

        case 2:      /* falls eine Einheit an dem EVENT-Knoten mit JEVNT = iCode = 2 eintrifft */
                     /* oder ein mit der Funktion SCHDL geplantes Ereignis vom Typ iEvent = iCode = 2 */
                     /* als aktuelles Ereignis laut Eintrag im Ereigniskalender von Visual SLAM eintritt */
                     /* Anweisungen */
                     break;
    }
}
```

Parameterbezeichnung	Erläuterung des zulässigen Argumentwertes
iCode	Benutzerdefinierter ganzzahliger Code, der mit dem aktuellen Ereignis verknüpft ist.
peUser	Zeiger, der auf diejenige Einheit zeigt, die das aktuelle Ereignis ausgelöst hat.

Abb. A-2.5.3.1: Allgemeine Form der benutzerspezifisierbaren C-Funktion „EVENT“ in Visual SLAM²⁷⁸

Die „EVENT“-Funktion dient jeweils dazu, die einem mit dem Ausdruck „iCode“ spezifizierten benutzerdefinierten Ereignis zugeordneten C-Programmanweisungen aufzurufen. Dies kann zum einen jeweils durch das Eintreffen einer Einheit an jenem EVENT-Knoten ausgelöst werden, dessen „JEVNT“-Spezifikation mit derjenigen von iCode übereinstimmt²⁷⁹. Zum anderen kann dies durch das Eintreten des mit Hilfe der Funktion SCHDL²⁸⁰ geplanten Ereignisses ausgelöst werden, dessen „iEvent“-Spezifikation ebenso mit der von „iCode“ überein-

²⁷⁷ Innerhalb der Programmiersprache C wird mit dem Begriff „Argument“ gemeinhin derjenige Wert bezeichnet, der bei einem Funktionsaufruf jeweils für die einzelnen Parameter einer zugrunde liegenden Funktion zu übergeben ist; vgl. dazu z.B. Kernighan, Ritchie 1990 /C/ S. 7.

²⁷⁸ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 389; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 544; Pritsker Corporation 1997 /Reference/ S. 124.

²⁷⁹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 544 f.; Pritsker Corporation 1997 /Reference/ S. 124.

²⁸⁰ Die „SCHDL“-Funktion ist eine benutzeraufrufbare C-Funktion, die im aktuellen Abschnitt unter (2) in (d) näher erläutert wird.

stimmt²⁸¹. Die unterschiedlichen „iCode“-spezifischen benutzerdefinierten Programmanweisungen werden mit Hilfe der switch-Anweisung der Programmiersprache C formuliert²⁸².

(b) „USERF“-Funktion

Allgemeine Form der Funktion USERF:

```
double SWFUNC USERF(int iCode, ENTITY * peUser)
{
    /* Vereinbarungen */
    switch (iCode)
    {
        case 1:      /* falls die mit iCode = 1 spezifizierte Funktion USERF aufgerufen wird */
                    /* Anweisungen */
                    break;

        case 2:      /* falls die mit iCode = 2 spezifizierte Funktion USERF aufgerufen wird */
                    /* Anweisungen */
                    break;
    }
}
```

Parameterbezeichnung	Erläuterung des zulässigen Argumentwertes
iCode	Benutzerdefinierter ganzzahliger Code der aufgerufenen benutzerspezifisierten Funktion USERF.
peUser	Zeiger, der auf diejenige Einheit zeigt, welche den aktuellen Funktionsaufruf ausgelöst hat.

Abb. A-2.5.3.2: Allgemeine Form der benutzerspezifisierbaren C-Funktion „USERF“ in Visual SLAM²⁸³

Die „USERF“-Funktion dient jeweils dazu, Variablenwerte mit Hilfe von „iCode“-spezifisch implementierten Programmanweisungen zu generieren und diese innerhalb der zugrunde liegenden Visual SLAM (Sub-) Netzwerke zuzuweisen²⁸⁴. Die „USERF“-Funktion wird dazu zuvorderst innerhalb der Spezifikation von ASSIGN-Knoten oder Aktivitäten angewendet²⁸⁵.

²⁸¹ Das Eintreten solcher Ereignisses wird in Visual SLAM mit Hilfe eines Ereigniskalenders gesteuert; vgl. Pitsker, O'Reilly 1999 /Visual SLAM/ S. 335.

²⁸² Mit der switch-Anweisung wird überprüft, ob ein Ausdruck (im oben angeführten Beispiel „iCode“) einen von mehreren ganzzahligen konstanten Werten aufweist. Trifft dies zu, so wird gemäß der Ausprägung von „iCode“ jeweils einleitend mit dem reservierten Wort „case“ und dem Wert von „iCode“ in Form sog. case-Marken "case 1", "case 2" usw. entsprechend verzweigt; vgl. Kernighan, Ritchie 1990 /C/ S. 58 f.

²⁸³ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 390; vgl. ferner hierzu Pitsker, O'Reilly 1999 /Visual SLAM/ S. 551 ff.; Pitsker Corporation 1997 /Reference/ S. 134 f.

²⁸⁴ Vgl. Pitsker, O'Reilly 1999 /Visual SLAM/ S. 551; Pitsker Corporation 1997 /Reference/ S. 134.

²⁸⁵ Vgl. Pitsker, O'Reilly 1999 /Visual SLAM/ S. 551. In ASSIGN-Knoten kann der Parameter „VALUE“ mithilfe der „USERF“-Funktion spezifiziert werden; vgl. dazu die Ausführungen zum ASSIGN-Knoten im Abschnitt A-2.3.3 in (2). In Aktivitäten kann der Parameter „Duration“ mithilfe der „USERF“-Funktion spezifiziert werden; vgl. dazu die Erläuterungen zu (NON-) SERVICE-Aktivitäten in Abschnitt A-2.3.2.

(2) Ausgewählte benutzeraufrufbare C-Funktionen

(a) „FILEM“-Funktion

Allgemeine Form der Funktion FILEM:

```
ENTRY* FILEM(UINT uiFile, ENTITY * peUser)
{
    ENTITY * peDateieinheit;      /* peDateieinheit wird als einheitenspezifischer Zeiger vereinbart. */
    peDateieinheit = su_entnew(0,NULL,NULL,NULL); /* Erzeugt eine neue Einheit (Einheitentyp 0). */
    FILEM(3,peDateieinheit);      /* Fügt die Einheit, auf die peDateieinheit zeigt, in die Datei mit der */
                                /* Datei-Nr. 3 ein. */
}
```

Parameterbezeichnung	Erläuterung des zulässigen Argumentwertes
uiFile	Benutzerdefinierter positiver ganzzahliger Code der aufgerufenen benutzeraufrufbaren Funktion FILEM.
peUser	Zeiger, der auf diejenige Einheit zeigt, die in die Datei mit der Nummer uiFile eingefügt werden soll.

Abb. A-2.5.3.3: Allgemeine Form der benutzeraufrufbaren C-Funktion „FILEM“ in Visual SLAM²⁸⁶

Die „FILEM“-Funktion zählt zu den benutzeraufrufbaren Funktionen zur Dateimanipulation²⁸⁷. Durch den Aufruf der „FILEM“-Funktion wird jeweils die Einheit, auf die mit dem Zeiger „peUser“ verwiesen wird, in diejenige Datei eingefügt, deren Dateinummer mit dem Wert von „uiFile“ übereinstimmt²⁸⁸.

²⁸⁶ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 394; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 549 f.; Pritsker Corporation 1997 /Reference/ S. 146.

²⁸⁷ Vgl. zu diesen Funktionen Pritsker Corporation 1997 /Reference/ S. 146 ff.

²⁸⁸ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 549.

(b) „SCHDL“-Funktion

Allgemeine Form der Funktion SCHDL:

```
ENTRY* SCHDL(int iEvent, ENTITY * peUser, double dTime )
{
    ENTITY * peKalendereinheit; /* peKalendereinheit wird als einheitenspezifischer Zeiger vereinbart. */

    peKalendereinheit = su_entnew(0,NULL,NULL,NULL); /* Erzeugt eine neue Einheit (Einheitentyp 0). */

    SCHDL(2,peKalendereinheit,3); /* Plant Ereignis (Ereignistyp 2) als zukünftiges Ereignis mit dem
                                   /* Ereigniszeitpunkt TNOW + 3 in die Warteschlangendatei mit der
                                   /* Datei-Nr. 0 ein. */
}
```

Parameterbezeichnung	Erläuterung des zulässigen Argumentwertes
iEvent	Benutzerdefinierter ganzzahliger Code der aufgerufenen benutzeraufrufbaren Funktion SCHDL.
peUser	Zeiger, der auf diejenige Einheit zeigt, die in die (Warteschlangen-)Datei des Ereigniskalenders (Datei-Nr. 0) eingefügt werden soll.
dTime	Reelle Variable mit nicht-negativem Wert.

Abb. A-2.5.3.4: Allgemeine Form der benutzeraufrufbaren C-Funktion „SCHDL“ in Visual SLAM²⁸⁹

Die „SCHDL“-Funktion zählt wie die zuvor unter (a) dargestellte Funktion „FILEM“ zu den Funktionen zur Dateimanipulation²⁹⁰. Die „SCHDL“-Funktion dient jeweils zur benutzerspezifischen Einplanung von Ereignissen in den Ereignis-Kalender von Visual SALM, welchem die Warteschlangendatei mit der Nr. „0“ exklusiv zugewiesen ist²⁹¹. Der jeweilige Ereigniszeitpunkt wird durch den Wert der Summe aus dem aktuellen Simulationszeitpunkt und dem im Parameter „dTime“ spezifizierten Wert determiniert. Die Ereignisse werden im Ereignis-Kalender jeweils durch eine Einheit repräsentiert. Die Rangposition jedes einzuplanenden Ereignisses wird dabei vom Visual SLAM Prozessor zugewiesen²⁹². Die „SCHDL“-Funktion liefert als Ergebnis den Platzhalter der jeweils zugewiesenen Rangposition²⁹³.

²⁸⁹ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 395; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 693 f.; Pritsker Corporation 1997 /Reference/ S. 147.

²⁹⁰ Vgl. zu diesen Funktionen Pritsker Corporation 1997 /Reference/ S. 146 ff.

²⁹¹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 693 f.; Pritsker Corporation 1997 /Reference/ S. 147.

²⁹² Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 335 ff.

²⁹³ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 693.

(c) „ENTER“-Funktion

Allgemeine Form der Funktion ENTER:

```

BOOL ENTER(UINT iCode, ENTITY * peUser)
{
    ENTITY * peDateieinheit;      /* peDateieinheit wird als einheitenspezifischer Zeiger vereinbart. */

    peDateieinheit = su_entnew(0,NULL,NULL,NULL); /* Erzeugt eine neue Einheit (Einheitentyp 0). */

    ENTER(2,peDateieinheit);      /* Fügt die Einheit, auf die peDateieinheit zeigt, an dem mit NUM = 2*/
                                /* spezifizierten ENTER-Knoten ein. */
}

```

Parameterbezeichnung	Erläuterung des zulässigen Argumentwertes
iCode	Benutzerdefinierter ganzzahliger Code der aufgerufenen benutzeraufrufbaren Funktion ENTER.
peUser	Zeiger, der auf diejenige Einheit zeigt, die an dem mit NUM = iCode spezifizierten ENTER-Knoten in das zugehörige Netzwerk eingefügt werden soll.

Abb. A-2.5.3.5: Allgemeine Form der benutzeraufrufbaren C-Funktion „ENTER“ in Visual SLAM²⁹⁴

Die „ENTER“-Funktion zählt zu den benutzeraufrufbaren Netzwerkstatusfunktionen²⁹⁵ und dient jeweils zum Einfügen derjenigen Einheit, auf die mit dem Zeiger „peUser“ verwiesen wird an demjenigen ENTER-Knoten, dessen Parameter „NUM“ mit dem Wert des Parameters „iCode“ spezifiziert ist²⁹⁶.

²⁹⁴ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 393; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 548; Pritsker Corporation 1997 /Reference/ S. 154.

²⁹⁵ Vgl. zu diesen Funktionen Pritsker Corporation 1997 /Reference/ S. 151 ff.

²⁹⁶ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 548; Pritsker Corporation 1997 /Reference/ S. 154.

(d) „su_entw“-Funktion

Allgemeine Form der Funktion su_entnew:

```
ENTITY* su_entnew(int iEntType, double * ATRIB, long * LTRIB, char ** STRIB)
{
    ENTITY * peDateieinheit;          /* peDateieinheit wird als einheitenspezifischer Zeiger vereinbart. */

    peDateieinheit = su_entnew(0,NULL,NULL,NULL);    /* Erzeugt eine neue Einheit (Einheitentyp 0). */
}
```

Parameterbezeichnung	Erläuterung des zulässigen Argumentwertes
iEntType	Benutzerdefinierter ganzzahliger Code der aufgerufenen benutzeraufrufbaren Funktion su_entnew zur Spezifikation eines Einheiten-Typs.
ATRIB	Zeiger, der auf das Feld (array) zeigt, in dem die reellen Attribute der zu generierenden Einheit verwaltet werden.
LTRIB	Zeiger, der auf das Feld (array) zeigt, in dem die ganzzahligen Attribute der zu generierenden Einheit verwaltet werden.
STRIB	Zeiger, der auf das Feld (array) zeigt, in dem die Zeichenketten-Attribute der zu generierenden Einheit verwaltet werden.

Abb. A-2.5.3.6: Allgemeine Form der benutzeraufrufbaren C-Funktion „su_entw“ in Visual SLAM²⁹⁷

Die „su_entnew“-Funktion zählt zu den einheitenspezifischen Statusfunktionen²⁹⁸ und dient jeweils zur Erzeugung einer (benutzerspezifischen) Einheit. Der Parameter „iEntType“ dient zur Spezifikation eines ganzzahligen Einheitentyps. Mit den Parametern „ATRIB“, „STRIB“ sowie „LTRIB“ werden Zeiger auf diejenigen Felder spezifiziert, in denen die zugehörigen initialisierten Attribute der betreffenden Einheit verwaltet werden²⁹⁹.

²⁹⁷ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 392; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 693 f.; Pritsker Corporation 1997 /Reference/ S. 160.

²⁹⁸ Vgl. zu diesen Funktionen Pritsker Corporation 1997 /Reference/ S. 159 ff.

²⁹⁹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 693; Pritsker Corporation 1997 /Reference/ S. 160.

A-2.6 Zu Steuerungsanweisungen in Visual SLAM

A-2.6.1 Steuerungsanweisungen in Steuerungsdateien

(1) „GEN“-Anweisung

<u>Programmanweisung in Visual SLAM:</u>		
GEN,"NAME","PROJECT",DATE,NNRNS,IXQT,IWARN,MXER;		
Eingabefeld	Option	Voreinstellung
NAME	Zeichenkette	unbesetzt
PROJECT	Zeichenkette	unbesetzt
DATE	Zeichenkette oder globaler Ausdruck	unbesetzt
NNRNS	positiver ganzzahliger globaler Ausdruck	unbesetzt (1 Simulationslauf)
IXQT	YES oder NO	YES
IWARN	YES oder NO	YES
MXERR	positiver ganzzahliger globaler Ausdruck	unbesetzt (1 Fehler)

Abb. A-2.6.1.1: „GEN“-Anweisung mit Parameterliste und Erläuterungen³⁰⁰

Mithilfe der „GEN“-Anweisung werden generelle Projektinformationen angelegt. Dabei werden in den Eingabefeldern (bzw. Parametern) „NAME“, „PROJECT“ und „DATE“ ein Projektverantwortlicher /-autor, eine Projektbezeichnung und ein Projektreferenzdatum (z.B. Projektbeginn) spezifiziert. Im Eingabefeld „NNRNS“ wird die Anzahl durchzuführender Simulationsläufe festgelegt. Mit dem Eingabefeld „IXQT“ wird determiniert, ob im Anschluss an eine erfolgreiche Übersetzung der Simulations-Eingabeanweisungen die Simulation ausgeführt werden soll. Mit dem Parameter „IWARN“ wird bestimmt, ob jeweils dann eine Fehlermeldung generiert werden soll, wenn während des Simulationsablaufs in einem (Sub-) Netzwerk eine Einheit unplanmäßig zerstört wird³⁰¹. Mit „MXERR“ wird die Anzahl der während der Simulationslaufzeit auftretenden Fehler bestimmt, die einen Abbruch des jeweiligen Simulationslaufs auslösen³⁰².

³⁰⁰ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 399; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 129 u. S. 341; Pritsker Corporation 1997 /Reference/ S. 60 f.

³⁰¹ Dabei werden Einheitenzerstörungen, die nicht mithilfe von TERMINATE-Knoten verursacht werden, als unplanmäßig bezeichnet. Zu solchen unplanmäßigen Einheitenzerstörungen kommt es z.B., wenn Einheiten, die aus einem Knoten weitergeleitet werden sollen, dessen ausströmende Aktivitäten allesamt mit Bedingungen spezifiziert sind, aufgrund (aktuell) fehlender Bedingungserfüllung von Visual SLAM zerstört wird.

³⁰² Vgl. Pritsker Corporation 1997 /Reference/ S. 60.

(2) „LIMITS“-Anweisung

<u>Programmanweisung in Visual SLAM:</u> LIMITS,MXX,MLL,MSZ,MATRIB,MLTRIB,MSTRIB,MNTRY;		
Eingabefeld	Option	Voreinstellung
MXX	ganzzahliger konstanter Ausdruck mit Wert >-2	-1 (keine XX-Variablen)
MLL	ganzzahliger konstanter Ausdruck mit Wert >-2	-1 (keine LL-Variablen)
MSZ	ganzzahliger konstanter Ausdruck mit Wert >-2	-1 (keine SZ-Variablen)
MATRIB	ganzzahliger konstanter Ausdruck mit Wert >-2	-1 (keine ATRIB-Variablen)
MLTRIB	ganzzahliger konstanter Ausdruck mit Wert >-2	-1 (keine LTRIB-Variablen)
MSTRIB	ganzzahliger konstanter Ausdruck mit Wert >-2	-1 (keine STRIB-Variablen)
MNTRY	positiver ganzzahliger konstanter Ausdruck	Leerzeichen (unendlich viele Einheiten)

Abb. A-2.6.1.2: „LIMITS“-Anweisung mit Parameterliste und Erläuterungen³⁰³

Die „LIMITS“-Anweisung dient einerseits zur Festlegung der maximal zulässigen Variablenanzahl der globalen Variablenarten („XX“, „LL“ und „SZ“) und der einheitenspezifischen Variablenarten („ATRIB“, „LTRIB“ und „STRIB“) auf der Hauptnetzwerkebene des zugrunde liegenden Szenarios mithilfe der Eingabefelder „MXX“, „MLL“ sowie „MSZ“ bzw. „MATRIB“, „MLTRIB“ und „MSTRIB“³⁰⁴ und andererseits zur Festlegung der maximalen Anzahl der im Verlauf der Simulation zeitgleich zu verwalteten Einheiten (Entities) mithilfe des Eingabefeldes „MNTRY“³⁰⁵.

³⁰³ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 400; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 129; Pritsker Corporation 1997 /Reference/ S. 62 f.

³⁰⁴ Für die globalen und einheitenspezifischen Variablen wird hierzu jeweils der maximale Index „I“ der zugehörigen Vektoren (z.B. „ATRIB“ oder „LL“) festgelegt.

³⁰⁵ Vgl. Pritsker Corporation 1997 /Reference/ S. 63.

(3) „NET“-Anweisung

<u>Programmanweisung in Visual SLAM:</u>		
NET,OPTION,FILE;		
Eingabefeld	Option	Voreinstellung
OPTION	READ, SAVE oder LOAD	READ
FILE	Zeichenkette	unbesetzt

Abb. A-2.6.1.3: „NET“-Anweisung mit Parameterliste und Erläuterungen³⁰⁶

Mithilfe der „NET“-Anweisung wird die Beschreibung der graphischen Netzwerkmodellierung des zu simulierenden Szenarios in Form entsprechender Visual SLAM-Programmanweisungen eingeleitet³⁰⁷. Dazu kann das Eingabefeld „OPTION“ entweder mit „READ“, „SAVE“ oder „LOAD“ spezifiziert werden. Wird „READ“ spezifiziert, wird dadurch die graphische Netzwerkbeschreibung für einen anstehenden Simulationslauf in Visual SLAM übersetzt. Bei einer Spezifikation mit „SAVE“ wird die graphische Netzwerkbeschreibung übersetzt und anschließend in derjenigen Datei gespeichert, deren Dateiname dazu im Eingabefeld „FILE“ zu spezifizierenden ist. Steht insofern in einer entsprechenden Datei eine übersetzte Netzwerkbeschreibung zur Verfügung, kann diese durch Spezifikation des Parameters „OPTION“ mit „LOAD“ und des betreffenden Dateinamens in „FILE“ zu entsprechenden Simulationen zwecken geladen werden³⁰⁸.

³⁰⁶ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 400; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 129 f.; Pritsker Corporation 1997 /Reference/ S. 65 f.

³⁰⁷ Vgl. Pritsker Corporation 1997 /Reference/ S. 65 f.

³⁰⁸ Vgl. Pritsker Corporation 1997 /Reference/ S. 66. Sofern im jeweils zugrunde liegenden Simulationsprojekt in den zu simulierenden Szenarios die Netzwerkmodellierung unverändert bleibt, sollte für das erste Szenario beim Eingabefeld „OPTION“ die Spezifikation „SAVE“ und für die weiteren Szenarios die „LOAD“-Spezifikation verwendet werden. Denn dadurch lässt sich die Rechenzeit, die erforderlich ist, um die Netzwerkbeschreibung zu übersetzen, ab dem zweiten zu simulierenden Szenario einsparen.

(4) „ARRAY“-Anweisung

<u>Programmanweisung in Visual SLAM:</u>		
ARRAY,IROW,NELEMENTS,{initial value, repeats};		
Eingabefeld	Option	Voreinstellung
IROW	eindeutige positive ganzzahlige Konstante	Angabe erforderlich
NELEMENTS	positive ganzzahlige Konstante	Angabe erforderlich
initial value	globaler Ausdruck	0

Abb. A-2.6.1.4: „ARRAY“-Anweisung mit Parameterliste und Erläuterungen³⁰⁹

Die „ARRAY“-Anweisung dient jeweils zur Definition und Initialisierung der in Visual SLAM (Sub-) Netzwerken in „ARRAYS“ (Felder) zur Verfügung stehenden Feldvariablen³¹⁰. Dabei wird mit „IROW“ die Feldnummer und mit „NELEMENTS“ die Anzahl der Feldvariablen definiert. Abweichend von dem voreingestellten Wert „0“ können die Feldvariablen benutzerspezifisch jeweils mit einem Initialwert (initial value) belegt werden³¹¹.

(5) „EQUIVALENCE“-Anweisung

<u>Programmanweisung in Visual SLAM:</u>		
EQUIVALENCE,{ {NAME,VAR},repeats};		
Eingabefeld	Option	Voreinstellung
NAME	Zeichenkette ohne Leerzeichen, die mit einem Buchstaben beginnt.	Angabe erforderlich
VAR	Zeichenkette (Variablenbezeichner)	Angabe erforderlich

Abb. A-2.6.1.5: „EQUIVALENCE“-Anweisung mit Parameterliste und Erläuterungen³¹²

Die „EQUIVALENCE“-Anweisung dient jeweils dazu, den jeweils im Eingabefeld „VAR“ anzugebenden einheitenspezifischen sowie globalen Variablen und Zufallsvariablen in Visual

³⁰⁹ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 402; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 97; Pritsker Corporation 1997 /Reference/ S. 55 f.

³¹⁰ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 97; Pritsker Corporation 1997 /Reference/ S. 55 f. Vgl. ferner hierzu die Erläuterung der Feldvariablen mit Feldnamen ARRAY in Abschnitt 4.2.1.3.1.

³¹¹ Vgl. Pritsker Corporation 1997 /Reference/ S. 55.

³¹² Entnommen aus Pütz 2004 /Produktionssysteme/ S. 402; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 96 f.; Pritsker Corporation 1997 /Reference/ S. 59.

SLAM im Eingabefeld „NAME“ einen eindeutigen sprachlichen Bezeichner bzw. Namen zuzuweisen, mit dem auf die zugehörige Variable referenziert werden kann³¹³. Dadurch wird insbesondere die Lesbarkeit bzw. das Verständnis der Simulationsmodellbestandteile erhöht und entsprechend die Handhabung der Simulationsmodelle vereinfacht³¹⁴.

(6) „INITIALIZE“-Anweisung

<u>Programmanweisung in Visual SLAM:</u>		
INITIALIZE,TTBEG,TTFIN,JJCLR,NCCLR,SICLR;		
Eingabefeld	Option	Voreinstellung
TTBEG	nicht negativer globaler Ausdruck	0.0
TTFIN	nicht negativer globaler Ausdruck > TTBEG	unendlich
JJCLR	YES oder NO	YES
NCCLR	ganzzahliger globaler Ausdruck	unbesetzt (JJCLR bezieht sich auf alle Datenerfassungsvariablen)
SICLR	YES oder NO	NO

Abb. A-2.6.1.6: „INITIALIZE“-Anweisung mit Parameterliste und Erläuterungen³¹⁵

Die „INITIALIZE“-Anweisung dient einerseits dazu, mit dem Eingabefeld „TTBEG“ den Simulationsstartzeitpunkt und mit dem Eingabefeld „TTFIN“ den Simulationsendzeitpunkt zu determinieren³¹⁶. Mit „JJCLR“ kann andererseits bestimmt werden, ob die während eines Simulationslaufs aufgrund der Spezifikation des Parameters „VALUE“ in COLCT-Knoten erfassten statistischen Daten vor Beginn durchzuführender Simulationsläufe jeweils gelöscht werden sollen³¹⁷. Mit dem Parameter „NCCLR“ kann dort spezifiziert werden, ab welchem COLCT-Knoten gemäß Index „N“ aufwärts die betreffenden statistischen Daten nicht gelöscht werden sollen. Mit dem Parameter „SICLR“ kann schließlich spezifiziert werden, ob OSN-Instanzen zwischen Simulationsläufen gelöscht werden sollen³¹⁸.

³¹³ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 96.

³¹⁴ Vgl. Pritsker Corporation 1997 /Reference/ S. 58.

³¹⁵ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 403; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 129 f.; Pritsker Corporation 1997 /Reference/ S. 61. Das oben angeführte Eingabefeld „SICLR“ (als Akronym für „subnetwork instance clear“) ist dem Dialogfeld „INITIALIZE Control“ des Control Builder von AweSim in der Version 3.0 entnommen.

³¹⁶ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 130; Pritsker Corporation 1997 /Reference/ S. 61.

³¹⁷ Vgl. hierzu auch die Erläuterungen zum COLCT-Knoten in Abschnitt 4.2.1.3.2.3, (16).

³¹⁸ Vgl. hierzu das Dialogfenster INITIALIZE Control des Control Builder von AweSim in der Version 3.0.

(7) „INTLC“-Anweisung

<u>Programmanweisung in Visual SLAM:</u> INTLC,{{VAR,VALUE},repeats};		
Eingabefeld	Option	Voreinstellung
VAR	Zeichenkette (globale Visual SLAM Variable)	Angabe erforderlich
VALUE	globaler Ausdruck oder Zeichenkette (Datentyp von VALUE und zugehörigere VAR müssen übereinstimmen)	Angabe erforderlich

Abb. A-2.6.1.7: „INTLC“-Anweisung mit Parameterliste und Erläuterungen³¹⁹

Die „INTLC“-Anweisung dient dazu, die im Eingabefeld „VAR“ jeweils anzugebenden globalen Visual SLAM Variablen benutzerspezifisch mit einem im Eingabefeld „VALUE“ anzugebenden Wert zu initialisieren³²⁰.

(8) „MONTR“-Anweisung

<u>Programmanweisung in Visual SLAM:</u> MONTR,OPTION,TFRST,TSEC,{TRACE Expression, repeats};		
Eingabefeld	Option	Voreinstellung
OPTION	SUMMARY, FILES, PROGRESS, STATES, CLEAR, TRACE, TRACE(NLBL, NLBL,...), ENTITY (ENUM, ENUM, ...), RESOURCE oder RESOURCE (RNUM, RNUM, ...)	Angabe erforderlich
TFRST	nicht-negativer globaler Ausdruck	0.0
TSECT	nicht-negativer globaler Ausdruck > TFRST	unendlich
TRACE Expression	beliebiger Ausdruck	unbesetzt

Abb. A-2.6.1.8: „MONTR“-Anweisung mit Parameterliste und Erläuterungen³²¹

Die MONTR-Anweisung dient jeweils zur Festlegung benutzerspezifischer Überwachungsaktionen in Simulationsläufen. Die einzelnen Überwachungsaktionen sind dazu ausgehend

³¹⁹ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 404; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 130; Pritsker Corporation 1997 /Reference/ S. 62.

³²⁰ Vgl. Pritsker Corporation 1997 /Reference/ S. 62. Mit jeder einzelnen INTLC-Anweisung kann eine Liste entsprechender Variablenwertinitialisierungen spezifiziert werden.

³²¹ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 404; vgl. ferner Pritsker Corporation 1997 /Reference/ S. 62.

vom Eingabefeld „OPTION“ zu spezifizieren³²². Die möglichen Überwachungsaktionsarten auf Basis der Spezifikation von „OPTION“ sind in der nachfolgenden Tabelle aufgelistet.

„OPTION“-Eingabefeld	Überwachungsaktion
SUMMARY	Ab dem in „TFRST“ spezifizierten Simulationszeitpunkt wird in mit „TSECT“ festgelegten Simulationszeitabständen jeweils ein Summary Report generiert.
FILES	Ab dem in „TFRST“ spezifizierten Simulationszeitpunkt wird in mit „TSECT“ festgelegten Simulationszeitabständen jeweils eine Ausgabe des Inhalts aller (Warteschlangen-) Dateien im Intermediate Report generiert.
PROGRESS	Ab dem in „TFRST“ spezifizierten Simulationszeitpunkt wird in mit „TSECT“ festgelegten Simulationszeitabständen während einer Simulation im Dialogfenster "AweSim! Simulator" bzw. im Unterfenster "SimulationsOutput" der Interactive Execution Environment ³²³ die aktuelle Simulationszeit eingeblendet.
STATES	Ab dem in „TFRST“ spezifizierten Simulationszeitpunkt wird in mit „TSECT“ festgelegten Simulationszeitabständen jeweils die Ausgabe der Werte der Variablen „SS[I]“ und „DD[I]“ im Intermediate Report generiert.
CLEAR	Ab dem in „TFRST“ spezifizierten Simulationszeitpunkt werden in mit „TSECT“ festgelegten Simulationszeitabständen die in diesen Zeiträumen von Visual SLAM erfassten statistischen Daten gelöscht. Diese Option dient dazu, etwaige in der Anlaufphase einer Simulation auftretenden (zufallsbedingten) Verzerrungen abzuwenden.
TRACE	In dem mit „TFRST“ und „TSECT“ spezifizierten Simulationszeitraum wird auf der Netzwerkebene ein Durchlaufprotokoll aller Einheiten erstellt, welches im Intermediate Report ausgegeben wird. Dabei wird jeweils festgehalten, zu welchem Simulationszeitpunkt die betreffenden Einheiten an welchem Knoten bzw. welcher Aktivität eingetroffen sind.
TRACE(NLBL1, NLBL2, ...)	Wie „TRACE“, wobei die Überwachungsaktion auf eine mit „NLBL1, NLBL2, ...“ benutzerspezifizierte Liste von Knoten beschränkt wird.
ENTITY	In dem mit „TFRST“ und „TSECT“ spezifizierten Simulationszeitraum werden auf der Netzwerkebene die Statusveränderungen sämtlicher Einheiten erfasst und im Intermediate Report ausgegeben.
ENTITY(ENUM1, ENUM2, ...)	Wie „ENTITY“, wobei die Überwachungsaktion auf diejenigen Einheiten beschränkt wird, deren Einheitennummer „ENUM“ in der Liste („ENUM1“, „ENUM2“, ...) aufgeführt ist.

³²² Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 343 f.; vgl. hierzu auch die Erläuterung der MONTR-Anweisung in SLAM II in Witte, Claus, Helling 1994 /SLAM/ S. 283 f.

³²³ Die Interactive Execution Environment wird in Abschnitt 4.2.1.7.3 erläutert.

„OPTION“-Eingabefeld	Überwachungsaktion
RESOURCE	In dem mit „TFRST“ und „TSECT“ spezifizierten Simulationszeitraum werden auf der Netzwerkebene die Statusveränderungen sämtlicher Ressourcen erfasst und im Intermediate Report ausgegeben.
RESOURCE(RNUM1, RNUM2, ...)	Wie „RESORUCE“, wobei die Überwachungsaktion auf diejenigen Ressourcen beschränkt wird, deren Ressourcennummer „RNUM“ in der Liste („RNUM1“, „RNUM2“, ...) aufgeführt ist.

Tab. A-2.6.1.1: Überwachungsaktionen auf Basis der Spezifikation des Eingabefeldes „OPTION“ innerhalb der „MONTR“-Anweisung³²⁴

Mit Ausnahme der Spezifikation des Eingabefeldes „OPTION“ entweder mit „CLEAR“ oder mit „PROGRESS“ ist die „MONTR“-Anweisung jeweils nur gültig für einen einzelnen Simulationslauf³²⁵.

(9) „PRIORITY“-Anweisung

<u>Programmanweisung in Visual SLAM:</u> PRIORITY,{ {IFILE,ranking},repeats};		
Eingabefeld	Option	Voreinstellung
IFILE	positiver ganzzahliger globaler Ausdruck oder CALENDAR	Angabe erforderlich
ranking	FIFO, LIFO, HVF(arg) oder LVF(arg)	FIFO

Abb. A-2.6.1.9: „PRIORITY“-Anweisung mit Parameterliste und Erläuterungen³²⁶

Mithilfe der „PRIORITY“-Anweisung wird jeweils die Abarbeitungsrangfolge bzw. -reihenfolge von Einheiten in einer oder mehreren im Eingabefeld „IFILE“ einzeln zu spezifizierenden (Warteschlangen-) Dateien im Eingabefeld „ranking“ festgelegt³²⁷. Abgesehen von der Ereignis-Warteschlangendatei³²⁸, welcher in Visual SLAM die Dateinummer „0“ exklusiv

³²⁴ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 378 in Anlehnung an Pritsker, O'Reilly 1999 /Visual SLAM/ S. 343 f.; Pritsker Corporation 1997 /Reference/ S. 64 f.; Witte, Claus, Helling 1994 /SLAM/ S. 284.

³²⁵ Vgl. hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 345.

³²⁶ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 406; vgl. ferner Pritsker, O'Reilly 1999 /Visual SLAM/ S. 342 f.; Pritsker Corporation 1997 /Reference/ S. 67.

³²⁷ Vgl. hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 130 f. u. S. 342 f.; Pritsker Corporation 1997 /Reference/ S. 67.

³²⁸ In der Ereignis-Warteschlangendatei ist in Visual SLAM der Ereigniskalender implementiert, der zur Verarbeitung und Steuerung der Simulationsereignisse benötigt wird. Auf die Ereignissteuerung in Visual SLAM wird in Abschnitt 3.2.2.2.1.2 unter (6) in (b) kurz eingegangen. Zu detaillierten Hinweisen zur Ereignis-

und fest zugewiesen ist, dient „FIFO“ im Sinne des „First-in-First-Out“-Prinzips als Voreinstellung der (übrigen) Warteschlangendateien.

Als primäres Kriterium der Abarbeitungsreihenfolge der Ereignis-Warteschlangendatei ist mit „LVF“ (Low Value First) der jeweils niedrigste Ereigniszeitpunkt der wartenden (Ereignisse repräsentierenden) Einheiten determiniert. In Visual SLAM werden die Ereignisse ausgehend von der aktuellen Simulationszeit „TNOW“ in aktuelle und zukünftige Ereignisse differenziert. Bei identischen Ereigniszeitpunkten innerhalb der aktuellen Ereignisse ist als sekundäres Abarbeitungsreihenfolgekriterium „LIFO“ (Last-in-First-Out) determiniert³²⁹. Dagegen ist bei identischen Ereigniszeitpunkten von zukünftigen Ereignissen „FIFO“ als sekundäres Abarbeitungsreihenfolgekriterium voreingestellt. Dieses sekundäre Abarbeitungsreihenfolgekriterium lässt sich mit Hilfe der „PRIORITY“-Anweisung durch Spezifikation des Parameters „ranking“ modifizieren³³⁰. Die Optionen dieses Parameters werden in der folgenden Tabelle erläutert.

„ranking“-Option	Erläuterung
FIFO	Die Einheiten werden in der Reihenfolgen ihres Eintreffens abgearbeitet. Diese Option kann mit Ausnahme der Ereignis-Warteschlangendatei (mit der Dateinummer „0“) uneingeschränkt für sämtliche (übrigen) Warteschlangendateien spezifiziert werden. Hinsichtlich der Ereignis-Warteschlangendatei kann „FIFO“ nur als sekundäres Abarbeitungskriterium zukünftiger Ereignisse spezifiziert werden.
LIFO	Die Einheiten werden bezüglich ihres Eintreffens in umgekehrter Reihenfolge abgearbeitet. Diese Option kann mit Ausnahme der Ereignis-Warteschlangendatei uneingeschränkt für sämtliche (übrigen) Warteschlangen-Dateien spezifiziert werden. Hinsichtlich der Ereignis-Warteschlangendatei kann „LIFO“ nur als sekundäres Abarbeitungskriterium für zukünftige Ereignisse spezifiziert werden.
HVF(arg)	Die Einheit mit dem höchsten Wert (High Value First) des beliebigen Ausdrucks „arg“ wird jeweils zunächst abgearbeitet. Diese Option kann mit Ausnahme der Ereignis-Warteschlangendatei uneingeschränkt für sämtliche (übrigen) Warteschlangen-Dateien als primäres Kriterium der Abarbeitungsreihenfolge spezifiziert werden; als sekundäres Kriterium gilt dabei zwingend FIFO. Hinsichtlich der Ereignis-Warteschlangendatei kann „HVF(arg)“ nur als sekundäres Abarbeitungskriterium zukünftiger Ereignisse spezifiziert werden.

verarbeitung und -steuerung wird verwiesen auf Pritsker, O'Reilly 1999 /Visual SLAM/ S. 335 ff. u S. 784 ff.

³²⁹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 338 f.

³³⁰ Vgl. Pritsker Corporation 1997 /Reference/ S. 67.

„ranking“-Option	Erläuterung
LVF(arg)	Die Einheit mit dem niedrigsten Wert (Low Value First) des beliebigen Ausdrucks „arg“ wird jeweils zunächst abgearbeitet. Diese Option kann mit Ausnahme der Ereignis-Warteschlangen-Datei uneingeschränkt für sämtliche (übrigen) Warteschlangen-Dateien als primäres Kriterium der Abarbeitungsreihenfolge spezifiziert werden; als sekundäres Kriterium gilt dabei zwingend FIFO. Hinsichtlich der Ereignis-Warteschlangen-Datei kann „LVF(arg)“ nur als sekundäres Abarbeitungskriterium zukünftiger Ereignisse spezifiziert werden.

Tab. A-2.6.1.2: Optionen des Parameters „ranking“ innerhalb der „PRIORITY“-Anweisung³³¹

(10) „RECORD“-Anweisung

<u>Programmanweisung in Visual SLAM:</u> RECORD, IPLOT, FILE, INDVAR, INDID, {FORMATs}, INDPF, TTSRT, TTEND, TTBT, {DEPVAR, DEPID, DEPPF}, repeats;		
Eingabefeld	Option	Voreinstellung
IPLOT	positiver ganzzahliger globaler Ausdruck (nur erforderlich, falls Daten durch Aufruf der GPLOT-Funktion generiert werden)	nächster sequentieller Index
FILE	Zeichenkette (Dateiname)	PLOTDATA.PLT
INDVAR	globaler Ausdruck (unabhängige Variable)	TNOW
INDID	Zeichenkette	"TIME"
FORMATs	AWESIM, TABLE, COMMA, EXCEL, DATABASE oder auch USER	AWESIM
INDPF	printf Format	Standardformat für INDVAR
TTSRT	nicht-negativer globaler Ausdruck	TTBEG (Simulationsstart)
TTEND	nicht-negativer globaler Ausdruck > TTSRT	TTFIN (Simulationsende)
TTBT	globaler Ausdruck	unbesetzt (Simulationszeitpunkt der Variablenänderung)
DEPVAR	globaler Ausdruck (abhängige Variable)	Angabe erforderlich
DEPID	Zeichenkette	unbesetzt
DEPPF	printf Format	Standardformat für DEPVAR

Abb. A-2.6.1.10: „RECORD“-Anweisung mit Parameterliste und Erläuterungen³³²

Die „RECORD“-Anweisung dient jeweils zur Generierung von benutzerspezifischen zweidimensionalen Plots (Trajektorien) oder auch Wertetabellen aus den jeweils in Simulationsläufen von AweSim gewonnenen Visual SLAM-Rohdaten, wobei für die entsprechende Daten-

³³¹ Vgl. hierzu die Ausführungen in 'PRIORITY Control Statement help' in der online-Hilfe des Control Builder von AweSim in der Version 3.0. Vgl. ferner hierzu Pritsker Corporation 1997 /Reference/ S. 67.

³³² Entnommen aus Pütz 2004 /Produktionssysteme/ S. 406; vgl. ferner Pritsker, O'Reilly 1999 /Visual SLAM/ S. 342 f.; Pritsker Corporation 1997 /Reference/ S. 67.

ausgabe unterschiedliche Ausgabeformate genutzt werden können³³³. Dazu sind jeweils eine unabhängige Variable im Eingabefeld „INDVAR“ sowie eine oder auch mehrere abhängige Variable(n) im Eingabefeld „DEPVAR“ zu spezifizieren. Sofern die betreffenden Rohdaten mit Hilfe der benutzeraufrufbaren Funktion „GPLOT“ gespeichert werden, ist der Parameter „IPLOT“ mit dem Wert (Nummer) des Parameters „uiPlot“ der Funktion „GPLOT“ entsprechend zu spezifizieren³³⁴. Die entsprechend erfassten Rohdaten können wahlweise in einem oder mehreren Format(en) (FORMATs), die in der nachfolgenden Tabelle aufgelistet sind, abgespeichert werden.

„FORMAT“-Option	Erläuterung
AWESIM	Speicherungsformat für AweSim-Plotdiagramme, die simulationsprojektspezifisch in der Menüoption „Graph“ des Report-Aktionsmenüs des AweSim-Anwendungsfensters aufgerufen werden können. Die Daten werden in einer Datei mit der Dateinamen-Erweiterung „plt“ gespeichert. Wird in „FILE“ die Voreinstellung beibehalten, setzt sich der Basisdateiname aus „Out“ und einer laufenden Nummer für jede RECORD-Anweisung zusammen (Dateinamenbeispiel: „Out1.plt“); andernfalls entspricht der Basisdateiname der gewählten „FILE“-Spezifikation.
TABLE	Die Datenausgabe erfolgt in tabellarischer Form in einer Datei mit der Dateinamen-Erweiterung „tab“. Der Basisdateiname wird analog zum „AWESIM“-Speicherformat vergeben.
COMMA	Die Datenausgabe erfolgt in kommaseparierter Form in einer Datei mit der Dateinamen-Erweiterung „cda“. Der Basisdateiname wird analog zum „AWESIM“-Speicherformat vergeben. Derartige Dateien können in diversen Tabellenkalkulations-Programmen importiert werden.
EXCEL	Die Datenausgabe erfolgt als Microsoft Excel 2.1-Tabelle in einer Datei mit der Dateinamen-Erweiterung „xls“. Der Basisdateiname wird analog zum „AWESIM“-Speicherformat vergeben.
DATABASE	Die Datenausgabe erfolgt in einem DBASE-Datenbankdateiformat in einer Datei mit der Dateinamen-Erweiterung „dbf“. Wird im Eingabefeld „FILE“ die Voreinstellung beibehalten, setzt sich der Basisdateiname aus „Raw“ und einer laufenden Nummer für jede RECORD-Anweisung zusammen (Dateinamenbeispiel: „Raw1.plt“); andernfalls entspricht der Basisdateiname der „FILE“-Spezifikation.

³³³ Vgl. hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 345; vgl. ferner die Erläuterungen zur RECORD-Anweisung der Simulationssprache SLAM II in Witte, Claus, Helling 1994 /SLAM/ S. 286.

³³⁴ Die benutzeraufrufbare C-Funktion „GPLOT“ [allgemeine Form: „BOOL GPLOT(UINT uiPlot)“] dient zur Speicherung der innerhalb der RECORD-Anweisung mit der Spezifikation von „IPLOT“= „uiPlot“ definierten Daten; vgl. dazu Pritsker Corporation 1997 /Reference/ S. 156.

„FORMAT“-Option	Erläuterung
USER	Die Datenausgabe erfolgt in einem benutzerdefinierten Dateiformat ³³⁵ in einer Datei mit einem benutzerspezifischen Dateinamen.

Tab. A-2.6.1.3: Optionen des Parameters „FORMATs“ innerhalb der „RECORD“-Anweisung³³⁶

Innerhalb der „RECORD“-Anweisungen kann jeweils mit dem Eingabefeld „FILE“ (wie in Tabelle A-2.6.1.3 erwähnt) ein Dateiname spezifiziert werden. Falls mit „FORMATs“ mehr als ein Speicherformat vereinbart wird, ist dazu lediglich der Basisdateiname in „FILE“ zu spezifizieren. Für jede unabhängige Variable kann im Eingabefeld „INDID“ und für die zugehörigen abhängigen Variablen können jeweils im Eingabefeld „DEPID“ ein sprachlicher Bezeichner zur Kennzeichnung der entsprechenden Spalten in den Wertetabellen spezifiziert werden. In Verbindung mit den Speicherformaten „TABLE“ sowie „COMMA“ können für unabhängige Variablen im Eingabefeld „INDPF“ und jeweils für die zugehörigen abhängigen Variablen im Eingabefeld „DEPPF“ Formatspezifikationen gemäß der C-Funktion „printf“ vereinbart werden³³⁷. Wird stattdessen die Voreinstellung von „INDPF“ bzw. „DEPPF“ beibehalten, wird von AweSim ein Standardausgabeformat erzeugt. Mit den Eingabefeldern „TTSRT“ und „TTEND“ können simulationslaufspezifisch der Start- und Endzeitpunkt der entsprechenden Rohdatenerfassung vereinbart werden. Mithilfe des Eingabefeldes „TTBET“ kann ein Intervall zwischen einzelnen Datenerfassungszeitpunkten determiniert werden. Sofern die Voreinstellung von „TTBET“ beibehalten wird, erfolgen zu jedem Simulationszeitpunkt, an dem Veränderungen der Variablenwerte der abhängigen Variablen auftreten, entsprechende Datenerfassungsmaßnahmen³³⁸.

³³⁵ Zur Generierung dieses benutzerdefinierten Dateiformats stehen in Visual SLAM für die in dieser Forschungsarbeit relevanten benutzerspezifischen Eingaben in Form von C-Programmen die benutzerspezifisierbaren C-Funktionen „sw_rawopen“, „sw_rawsafe“ und „sw_rawend“ zur Verfügung; dazu sei verwiesen auf Pritsker Corporation 1997 /Reference/ S. 68 u. S. 137 f.

³³⁶ Weitestgehend entnommen aus Pütz 2004 /Produktionssysteme/ S. 409 in Anlehnung an Pritsker Corporation 1997 /Reference/ S. 68 f.

³³⁷ Vgl. Pritsker Corporation 1997 /Reference/ S. 68.

³³⁸ Vgl. Pritsker Corporation 1997 /Reference/ S. 69.

(11) „SEEDS“-Anweisung

<u>Programmanweisung in Visual SLAM:</u>		
SEEDS,{{ISEED,IS,R}, repeats};		
Eingabefeld	Option	Voreinstellung
ISEED	ungerader ganzzahliger globaler Ausdruck	intern von Visual SLAM zugewiesene Konstante
IS	positiver ganzzahliger konstanter Ausdruck im Intervall [1, 100]	nächster sequentieller Zufallszahlenstrom
R	YES oder NO	NO

Abb. A-2.6.1.11: „SEEDS“-Anweisung mit Parameterliste und Erläuterungen³³⁹

Die „SEEDS“-Anweisung dient zur Initialisierung benutzerspezifischer Startwerte, die für einen der insgesamt 100 in Visual SLAM verfügbaren Pseudozufallszahlenströme „IS“ im Eingabefeld „ISEED“³⁴⁰ festzulegen sind. Falls mehrere unmittelbar aufeinander folgende Simulationsläufe realisiert werden sollen, kann zudem mit dem Eingabefeld „R“ festgelegt werden, ob der jeweilige Zufallszahlenstrom vor jedem einzelnen Simulationslauf reinitialisiert werden soll³⁴¹.

³³⁹ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 410; vgl. ferner Pritsker, O'Reilly 1999 /Visual SLAM/ S. 343; Pritsker Corporation 1997 /Reference/ S. 70 f.

³⁴⁰ Eine Reihe von Methoden zur Generierung von Pseudozufallszahlen, wie z.B. die Alias-Methode zur Generierung diskret verteilter Pseudozufallszahlen sowie die Inversions- und die Verwerfungsmethode zur Generierung stetig verteilter Pseudozufallszahlen, führen in Simulationen, bei denen mithilfe von Zufallszahlen (als Inputvariablen) Erwartungswerte von Outputvariablen geschätzt werden, bei jeweils relativ geringer Anzahl durchgeführter Simulationsläufe verstärkt zum Problem hoher Varianzen der Mittelwerte der Zufallsvariablen (welche als Schätzwerte für die betreffenden Erwartungswerte fungieren). Daher bedarf es häufig einer Vielzahl von Simulationsläufen (häufig 10000 und sogar mehr) zur Reduzierung der betreffenden Varianzen. Der Varianzreduzierung dienen u.a. die Methode der komplementären Zufallszahlen und die Methode der sog. antithetischen Zufallszahlen, die eine Verallgemeinerung der Methode der komplementären Zufallszahlen darstellt; vgl. dazu z.B. Neumann, Morlock 1993 /Research/ S. 705 ff. u. 725 ff. inklusive der dort angegebenen weiterführenden Literaturhinweise. Wird der Parameter „ISEED“ jeweils mit einem negativen Wert belegt, werden aufgrund dessen in Visual SLAM jeweils antithetische Pseudozufallszahlen erzeugt; vgl. dazu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 343 u. S. 465 ff.

³⁴¹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 343; Pritsker Corporation 1997 /Reference/ S. 70 f.

(12) „ENTRY“-Anweisung

<u>Programmanweisung in Visual SLAM:</u>		
ENTRY,IFILE,NLBL,{attribute value, repeats};		
Eingabefeld	Option	Voreinstellung
IFILE	positive ganzzahlige Konstante	unbesetzt
NLBL	Zeichenkette (Knotenbezeichner)	unbesetzt (IFILE bestimmt den Knoten)
attribute value	komma separierte Initialwerte in der Reihenfolge: reelle globale Ausdrücke, gefolgt von ganzzahligen globalen Ausdrücken und Zeichenketten	0.0, 0 oder " "

Abb. A-2.6.1.12: „ENTRY“-Anweisung mit Parameterliste und Erläuterungen³⁴²

Die „ENTRY“-Anweisung dient jeweils dazu, eine Einheit zu Beginn eines Simulationslaufs entweder in die im Eingabefeld „IFILE“ zu spezifizierende Datei einzufügen oder zu demjenigen Knoten in ein (Sub-) Netzwerk zu leiten, dessen Knotenbezeichner im Eingabefeld „NLBL“ anzugeben ist³⁴³. Die einheitenspezifischen Variablen der betreffenden Einheit können abweichend von der Vorbelegung {ATRI[B] mit 0.0, LTRIB[I] mit 0 und STRIB[I] mit " "} mit benutzerspezifisierten Initialwerten belegt werden. Dabei ist die Reihenfolge zu beachten, wonach zunächst ATRIB[I], dann LTRIB[I] und schließlich STRIB[I] zu initialisieren sind. Ist das Eingabefeld „IFILE“ mit einer Dateinummer spezifiziert, welche sich entweder auf einen QUEUE- oder einen AWAIT-Knoten (mit Knotenbezeichner „NLBL“) bezieht, führt die „ENTRY“-Anweisung dazu, dass die zugrunde liegende Einheit zu Beginn der Simulation an dem entsprechenden QUEUE- bzw. AWAIT-Knoten eintrifft. „ENTRY“-Anweisungen wirken sich jeweils nur auf einzelne Simulationsläufe aus³⁴⁴.

³⁴² Entnommen aus Pütz 2004 /Produktionssysteme/ S. 610; vgl. ferner Pritsker, O'Reilly 1999 /Visual SLAM/ S. 129 f.; Pritsker Corporation 1997 /Reference/ S. 58.

³⁴³ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 130.

³⁴⁴ Vgl. Pritsker Corporation 1997 /Reference/ S. 58.

(13) „SEVNT“-Anweisung

<u>Programmanweisung in Visual SLAM:</u>		
SEVNT,JEVNT,XVAR,XDIR,VALUE,XTOL;		
Eingabefeld	Option	Voreinstellung
JEVNT	ganzzahliger globaler Ausdruck	Angabe erforderlich
XVAR	globaler Ausdruck	Angabe erforderlich
XDIR	POSITIVE,NEGATIVE oder EITHER	EITHER
VALUE	globaler Ausdruck	Angabe erforderlich
XTOL	nicht-negativer konstanter Ausdruck	unendlich

Abb. A-2.6.1.13: „SEVNT“-Anweisung mit Parameterliste und Erläuterungen³⁴⁵

Die „SEVNT“-Anweisung dient jeweils dazu, diejenige „EVENT“-Funktion aufzurufen, deren Parameter „iCode“ mit dem Wert der Spezifikation des Eingabefeldes „JEVNT“ übereinstimmt. Der Aufruf der betreffenden „EVENT“-Funktion erfolgt, sobald der im Eingabefeld „XVALUE“ zu spezifizierende Schwellenwert des im Eingabefeld „XVARW zu spezifizierenden globalen Ausdrucks gemäß der im Eingabefeld „XDIR“ Wertänderungsrichtung (negativ und oder positiv) gekreuzt wird³⁴⁶. Für das entsprechende Kreuzen des Schwellenwertes kann im Eingabefeld „XTOL“ ein Toleranzwert festgelegt werden, sofern „XVAR“ mit Statusvariablen „SS“ oder deren Ableitungen „DD“ (im Rahmen einer kontinuierlichen Simulation) spezifiziert ist³⁴⁷.

³⁴⁵ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 613; vgl. ferner Pritsker, O'Reilly 1999 /Visual SLAM/ S. 737; Pritsker Corporation 1997 /Reference/ S. 71.

³⁴⁶ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 736 ff.; Pritsker Corporation 1997 /Reference/ S. 71.

³⁴⁷ Vgl. Pritsker Corporation 1997 /Reference/ S. 71.

(14) „TIMST“-Anweisung

<u>Programmanweisung in Visual SLAM:</u>		
TIMST,ISTAT,VAR,ID,NCEL,HLOW,HWID;		
Eingabefeld	Option	Voreinstellung
ISTAT	eindeutiger positiver ganzzahliger globaler Ausdruck	nächster sequentieller Index
VAR	Globaler Ausdruck	Angabe erforderlich
ID	Zeichenkette	unbesetzt
NCEL	nicht-negativer ganzzahliger globaler Ausdruck	0 (kein Histogramm)
HLOW	globaler Ausdruck	0.0
HWID	positiver globaler Ausdruck	1.0

Abb. A-2.6.1.14: „TIMST“-Anweisung mit Parameterliste und Erläuterungen³⁴⁸

Die „TIMST“-Anweisung dient jeweils dazu, die Entwicklung eines im Eingabefeld „VAR“ zu spezifizierenden globalen Ausdrucks während eines Simulationslaufs permanent zu erfassen. Für den TIME-PERSISTANT STATISTIK REPORT kann jede „TIMST“-Anweisung durch Spezifikation des Eingabefeldes „ID“ mit einem sprachlichen Bezeichner gekennzeichnet werden. Darüber hinaus kann jede TIMST-Anweisung im Eingabefeld „ISTAT“ mit einer eindeutigen Kennziffer ausgestattet werden. Aus den mithilfe von „TIMST“-Anweisungen erfassten Werten können Histogramm-Informationen generiert werden. Dazu sind in den Eingabefeldern „NCEL“, „HLOW“, „HWID“ der „TIMST“-Anweisung die Klassenanzahl, die oberste Grenze der untersten Klasse sowie die Klassenbreite zu spezifizieren³⁴⁹.

³⁴⁸ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 613; vgl. ferner Pritsker, O'Reilly 1999 /Visual SLAM/ S. 736; Pritsker Corporation 1997 /Reference/ S. 72.

³⁴⁹ Vgl. Pritsker Corporation 1997 /Reference/ S. 72.

(15) „DOCOLCT“-Anweisung

<u>Programmanweisung in Visual SLAM:</u>		
DOCOLCT, COLLECT, "TABLE", "VARIABLE";		
Eingabefeld	Option	Voreinstellung
COLLECT	positiver ganzzahliger globaler Ausdruck	Angabe erforderlich
"TABLE"	Zeichenkette	unbesetzt
"VARIABLE"	Zeichenkette	"TABLE"-Spezifikation

Abb. A-2.6.1.15: „DOCOLCT“-Anweisung mit Parameterliste und Erläuterungen³⁵⁰

Die „DOCOLCT“-Anweisung dient jeweils zur Speicherung von Werten, die mithilfe des COLCT-Knoten erfasst werden, in eine Datenbankdatei³⁵¹. Dazu ist mit dem Eingabefeld „COLLECT“ der Indexwert des COLCT-Knotens gemäß der Spezifikation des zugehörigen Parameters „N“ festzulegen. Im Eingabefeld „TABLE“ ist der Dateiname der betreffenden Datenbankdatei anzugeben und im Eingabefeld „Variable“ ist der Spaltenname derjenigen Spalte der Datenbankdatei zu spezifizieren, in der die erfassten Werte abgespeichert werden sollen³⁵².

³⁵⁰ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 614; vgl. ferner Pritsker, O'Reilly 1999 /Visual SLAM/ S. 345; Pritsker Corporation 1997 /Reference/ S. 57.

³⁵¹ Vgl. Pritsker, O'Reilly 1999 /Visual SLAM/ S. 345.

³⁵² Vgl. Pritsker Corporation 1997 /Reference/ S. 57.

A-2.6.2 Die „LIMITSVSN“-Anweisung in objektorientierten Subnetzwerkklassen

Programmanweisung in Visual SLAM: LIMITSVSN,MXXINST,MLLINST,MSZINST,MANTRIB,MLNTRIB,MSNTRIB;		
Eingabefeld	Option	Voreinstellung
MXXINST	ganzzahliger globaler Ausdruck mit Wert > -2	-1 (keine XXINST-Variablen)
MLLINST	ganzzahliger globaler Ausdruck mit Wert > -2	-1 (keine LINST-Variablen)
MSZINST	ganzzahliger globaler Ausdruck mit Wert > -2	-1 (keine SZINST-Variablen)
MANTRIB	ganzzahliger globaler Ausdruck mit Wert > -2	-1 (keine ANTRIB-Variablen)
MLNTRIB	ganzzahliger globaler Ausdruck mit Wert > -2	-1 (keine LNTRIB-Variablen)
MSNTRIB	ganzzahliger globaler Ausdruck mit Wert > -2	-1 (keine SNTRIB-Variablen)

Abb. A-2.6.2: „LIMITSVSN“-Anweisung mit Parameterliste und Erläuterungen³⁵³

Analog zum Anwendungszweck der „LIMITS“-Anweisung innerhalb von Steuerungsdateien dient die „LIMITSVSN“-Anweisung³⁵⁴ in OSN-Klassen dazu, die maximale Variablenanzahl der einzelnen lokalen benutzerspezifischen Variablenarten „XXINST“, „LLINST“, „SZINST“ mit den entsprechenden Parametern „MXXINST“, „MLLINST“ und „MSZINST“ sowie die maximale Variablenanzahl der einzelnen einheitenspezifischen Variablenarten „ANTRIB“, „LNTRIB“ und „SNTRIB“ mit den zugehörigen Parameter „MANTRIB“, „MLNTRIB“ und „MSNTRIB“ festzulegen³⁵⁵.

³⁵³ Entnommen aus Pütz 2004 /Produktionssysteme/ S. 412; vgl. ferner hierzu Pritsker, O'Reilly 1999 /Visual SLAM/ S. 129; Pritsker Corporation 1997 /Reference/ S. 62 f.

³⁵⁴ Hinsichtlich der „LIMITS“-Anweisung sei verwiesen auf die entsprechenden Ausführungen oben in Abschnitt A-2.6.1 in (2)

³⁵⁵ Vgl. Pritsker Corporation 1997 /Reference/ S. 62 f.

A-3 Anhang zum Simulationsprojekt MCIPEC

A-3.1 Liste der im Simulationsprojekt MCIPEC enthaltenen Visual SLAM-Variablen (ohne Berücksichtigung von Feldvariablen)

Lfd.-Nr.	Variablenbezeichner	Visual SLAM-Variable	Initialwert	Erläuterung
1	AktAnzNegVeria_AktProz	LTRIB[3]		Ganzzahlige einheitenpsepezifische Variable, die zur Verwaltung der aktuellen Anzahl negativer Verifikationsergebnisse (vom Typ Veri_a) dient, die für den aktuellen Prozess vorliegen.
2	AktAnzNegVeria10i2j33s0u0	LL[1]	0	Aktuelle Anzahl negativer Verifikationsergebnisse (vom Typ Veri_a), die für den Prozess 10i2j33s0u0 vorliegen.
3	AktValiP_Bppzl	STRIB[4]		Einheitenspezifische Zeichenkettenvariable, die zur Angabe derjenigen Liste Projektzielen zugeordneter Beitragspositionen dient, die aktuell für eine Validitätsprüfung herangezogen wird.
4	AktValiP_PB_BppzlARRAY	LTRIB[5]		Ganzzahlige einheitenspezifische Variable, die zur Angabe der Nr. desjenigen ARRAYS dient, in dem diejenige Liste mit geplanten Beiträgen zur Erreichung von Projekzielen verwaltet wird, die aktuell zur Validitätsprüfung herangezogen wird.
5	AktValiPErgebnis	STRIB[6]		Einheitenspezifische Zeichenkettenvariable, die zur Angabe des Ergebnisses der aktuellen Validitätsprüfung herangezogen wird; mögliche Zeichenketten (gemäß SNr. 385): "TRUE", "PartialFALSE" und "TotalFALSE".
6	AktVeriVAvinr	LTRIB[1]		Ganzzahlige einheitenspezifische Variable, die zur Angabe der Nr. derjenigen Anwendungsvoraussetzungsliste dient, die aktuell für einen Verifikationsversuch herangezogen wird.
7	AktVeriVERgebnis	STRIB[3]		Einheitenspezifische Zeichenkettenvariable, die zur Angabe des Ergebnisses des aktuellen Verifikationsversuchs herangezogen wird; mögliche Zeichenketten (gemäß SNr. 319): "TotalTRUE", "TotalFALSE", "PartialTRUE" und "PartialFALSE".
8	AktVeriVersuchAvl	STRIB[1]		Einheitenspezifische Zeichenkettenvariable, die zur Angabe derjenigen Anwendungsvoraussetzungsliste dient, die aktuell für einen Verifikationsversuch herangezogen wird.
9	AktVeriVersuchPRegel	STRIB[2]		Einheitenspezifische Zeichenkettenvariable, die zur Angabe derjenigen (Prozess-) Regel dient, die aktuell für einen Verifikationsversuch herangezogen wird.

Lfd.-Nr.	Variablenbezeichner	Visual SLAM-Variable	Initialwert	Erläuterung
10	AkValiP_PRegel	STRIB[5]		Einheitenspezifische Zeichenkettenvariable, die zur Angabe derjenigen (Prozess-) Regel dient, die aktuell für eine Validitätsprüfung herangezogen wird.
11	AnzAvAktAvl	LTRIB[2]		Ganzzahlige einheitenspezifische Variable, die zur Angabe der Anzahl der Anwendungsvoraussetzungen dient, die in der Anwendungsvoraussetzungsliste enthalten ist, die aktuell für einen Verifikationsversuch herangezogen wird.
12	AnzProjektziele	LL[36]	4	Globale ganzzahlige Variable, die zur Angabe der Anzahl der im zugrunde liegenden Projekt verfolgten Projektziele dient.
13	bErfuellungAv30	LL[30]	0	Globale (boolsche) Variable zur Angabe der Erfüllung (Wert 1) oder Nichterfüllung (Wert 0) der Anwendungsvoraussetzung Av30
14	bErfuellungAv31	LL[31]	0	Globale (boolsche) Variable zur Angabe der Erfüllung (Wert 1) oder Nichterfüllung (Wert 0) der Anwendungsvoraussetzung Av31
15	bErfuellungAv32	LL[32]	0	Globale (boolsche) Variable zur Angabe der Erfüllung (Wert 1) oder Nichterfüllung (Wert 0) der Anwendungsvoraussetzung Av32
16	bErfuellungAv33	LL[33]	0	Globale (boolsche) Variable zur Angabe der Erfüllung (Wert 1) oder Nichterfüllung (Wert 0) der Anwendungsvoraussetzung Av33
17	bErfuellungAv34	LL[34]	0	Globale (boolsche) Variable zur Angabe der Erfüllung (Wert 1) oder Nichterfüllung (Wert 0) der Anwendungsvoraussetzung Av34
18	bErfuellungAv35	LL[35]	0	Globale (boolsche) Variable zur Angabe der Erfüllung (Wert 1) oder Nichterfüllung (Wert 0) der Anwendungsvoraussetzung Av35
19	KA_Av30	SZ[30]	"K"	Globale Zeichenkettenvariable zur Angabe der Kritizitätsausprägung der Anwendungsvoraussetzung AV30; mögliche Zeichen: "K" für kritisch und "U" für unkritisch.
20	KA_Av31	SZ[31]	"K"	Globale Zeichenkettenvariable zur Angabe der Kritizitätsausprägung der Anwendungsvoraussetzung AV31; mögliche Zeichen: "K" für kritisch und "U" für unkritisch.
21	KA_Av32	SZ[32]	"K"	Globale Zeichenkettenvariable zur Angabe der Kritizitätsausprägung der Anwendungsvoraussetzung AV32; mögliche Zeichen: "K" für kritisch und "U" für unkritisch.

Lfd.-Nr.	Variablenbezeichner	Visual SLAM-Variable	Initialwert	Erläuterung
22	KA_Av33	SZ[33]	"K"	Globale Zeichenkettenvariable zur Angabe der Kritizitätsausprägung der Anwendungsvoraussetzung AV33; mögliche Zeichen: "K" für kritisch und "U" für unkritisch.
23	KA_Av34	SZ[34]	"K"	Globale Zeichenkettenvariable zur Angabe der Kritizitätsausprägung der Anwendungsvoraussetzung AV34; mögliche Zeichen: "K" für kritisch und "U" für unkritisch.
24	KA_Av35	SZ[35]	"U"	Globale Zeichenkettenvariable zur Angabe der Kritizitätsausprägung der Anwendungsvoraussetzung AV35; mögliche Zeichen: "K" für kritisch und "U" für unkritisch.
25	Knv_AktProz	LTRIB[4]		Einheitenspezifische Variable, die zur Angabe der kritischen Anzahl negativer Verifikationsversuche des aktuell zur Verifikation herangezogenen Prozesses dient
26	Knv10i2j33s0u0	LL[2]	4	Globale ganzzahlige Variable, die zur Angabe der kritischen Anzahl negativer Verifikationsversuche des Prozesses 10i2j33s0u0 dient.
27	LokGvon10i2j32s0u0zu10i2j33s0u0	XX[1]		Lokales Gewicht des Prozesses 10i2j33s0u0 ausgehend vom Prozess 10i2j32s0u0
28	LokGvon10i2j32s0u0zu10i2j34s0u0	XX[2]		Lokales Gewicht des Prozesses 10i2j34s0u0 ausgehend vom Prozess 10i2j32s0u0
29	LokGvon10i2j32s0u0zu10i2j36s0u0	XX[3]		Lokales Gewicht des Prozesses 10i2j36s0u0 ausgehend vom Prozess 10i2j32s0u0
30	LokGvon10i2j35s0u0zu10i2j38s0u0	XX[5]		Lokales Gewicht des Prozesses 10i2j38s0u0 ausgehend vom Prozess 10i2j35s0u0
31	LokGvon10i2j35s0u0zu10i2j37s0u0	XX[4]		Lokales Gewicht des Prozesses 10i2j37s0u0 ausgehend vom Prozess 10i2j35s0u0
32	PB_Bppzl233	LL[37]	102330	Globale ganzzahlige Variable, die zur Angabe der Nr. des ARRAYS dient, in dem die geplanten Beiträge zur Erreichung von Projekzielen bzgl. der Liste Bppzl233 verwaltet werden.
33	SB_Bppzl233	LL[38]	102332	Globale ganzzahlige Variable, die zur Angabe der Nr. des ARRAYS dient, in dem die simulierten Beiträge zur Erreichung von Projekzielen bzgl. der Liste Bppzl233 verwaltet werden.

Lfd.-Nr.	Variablenbezeichner	Visual SLAM-Variable	Initialwert	Erläuterung
34	SWA_Bppzl233	LL[39]	102333	Globale ganzzahlige Variable, die zur Angabe der Nr. des ARRAYs dient, in dem Soll-Wird-Abweichungswerte bzgl. der Liste Bppzl233 verwaltet werden.

Tab. A-3.1: Liste der im Simulationsprojekt MCIPEC enthaltenen Visual SLAM-Variablen
(ohne Berücksichtigung von Feldvariablen)

A-3.2 Steuerungsanweisungen der Steuerungsdatei CVER_TP2

Zur Strukturierung der nachfolgend aufgelisteten Steuerungsanweisungen der Steuerungsdatei CVER_TP2 sind zum einen die obligatorischen Steuerungsanweisungen GEN, LIMITS, NET sowie FIN und zum anderen die Kommentarzeilen, die jeweils mit einem Semikolon eingeleitet sind, fett hervorgehoben. Ferner sind die Steuerungsanweisungen, welche über eine Ausgabezeile hinausgehen, jeweils ab der zweiten Zeile eingerückt.

Die Steuerungsdatei CVER_TP2 enthält folgende Steuerungsanweisungen:

```
GEN,"Dr. Markus Puetz","MCIPEC Demobeispiel",17.09.2009,1,YES,YES;
LIMITS,3,36,36,,3,3,100;
;Sprachliche Bezeichner fuer globale Variablen:
EQUIVALENCE,{ {AktAnzNegVeria10i2j33s0u0,LL[1]} };
EQUIVALENCE,{ {bErfuellungAv30,LL[30]}, {bErfuellungAv31,LL[31]},
    {bErfuellungAv32,LL[32]} };
EQUIVALENCE,{ {bErfuellungAv33,LL[33]}, {bErfuellungAv34,LL[34]},
    {bErfuellungAv35,LL[35]} };
EQUIVALENCE,{ {KA_Av30,SZ[30]}, {KA_Av31,SZ[31]}, {KA_Av32,SZ[32]},
    {KA_Av33,SZ[33]} };
EQUIVALENCE,{ {KA_Av34,SZ[34]}, {KA_Av35,SZ[35]} };
EQUIVALENCE,{ {LokGvon10i2j32s0u0zu10i2j33s0u0,XX[1]},
    {LokGvon10i2j32s0u0zu10i2j34s0u0,XX[2]} };
;Sprachliche Bezeichner fuer einheitenspezifische Variablen:
EQUIVALENCE,{ {AktVeriVAvlNr,LTRIB[1]}, {AnzAvAktAvl,LTRIB[2]},
    {AktAnzNegVeria_AktProz,LTRIB[3]} };
EQUIVALENCE,{ {AktVeriVersuchAvl,STRIB[1]}, {AktVeriVersuchPRegel,STRIB[2]},
    {AktVeriVERgebnis,STRIB[3]} };
;Anwendungsvoraussetzungsliste Avl233:
ARRAY,233,6,{30,31,32,33,34,35};
;Initialisierung globaler Variablen:
INTLC,{ {AktAnzNegVeria10i2j33s0u0,0} };
INTLC,{ {bErfuellungAv30,0}, {bErfuellungAv31,0}, {bErfuellungAv32,0},
    {bErfuellungAv33,0} };
INTLC,{ {bErfuellungAv34,0}, {bErfuellungAv35,0} };
INTLC,{ {KA_Av30,"K"}, {KA_Av31,"K"}, {KA_Av32,"K"}, {KA_Av33,"K"},
    {KA_Av34,"K"}, {KA_Av35,"U"} };
;Simulationszeitraum: tp2:= 20880 bis tm1:= 526320
INITIALIZE,20880,526320,YES,,NO;
NET;
FIN;
```

A-3.3 Steuerungsanweisungen der Steuerungsdatei CVER_TP6

Die nachfolgend aufgelisteten Steuerungsanweisungen der Steuerungsdatei CVER_TP6 sind in der Weise strukturiert, die bereits für die Auflistung der Steuerungsanweisungen der Steuerungsdatei CVER_TP2 zugrunde gelegt wurde³⁵⁶.

Die Steuerungsdatei CVER_TP6 enthält folgende Steuerungsanweisungen:

```

GEN,"Dr. Markus Puetz","MCIPEC Demobeispiel",17.09.2009,1,YES,YES;
LIMITS,3,36,36,,4,3,100;
;Sprachliche Bezeichner fuer globale Variablen:
EQUIVALENCE,{ {AktAnzNegVeria10i2j33s0u0,LL[1]} };
EQUIVALENCE,{ {bErfuellungAv30,LL[30]}, {bErfuellungAv31,LL[31]},
  {bErfuellungAv32,LL[32]} };
EQUIVALENCE,{ {bErfuellungAv33,LL[33]}, {bErfuellungAv34,LL[34]},
  {bErfuellungAv35,LL[35]} };
EQUIVALENCE,{ {KA_Av30,SZ[30]}, {KA_Av31,SZ[31]}, {KA_Av32,SZ[32]},
  {KA_Av33,SZ[33]} };
EQUIVALENCE,{ {KA_Av34,SZ[34]}, {KA_Av35,SZ[35]} };
EQUIVALENCE,{ {Knv10i2j33s0u0,LL[2]} };
EQUIVALENCE,{ {LokGvon10i2j32s0u0zu10i2j33s0u0,XX[1]},
  {LokGvon10i2j32s0u0zu10i2j34s0u0,XX[2]} };
;Sprachliche Bezeichner fuer einheitenspezifische Variablen:
EQUIVALENCE,{ {AktVeriVAvlNr,LTRIB[1]}, {AnzAvAktAvl,LTRIB[2]},
  {AktAnzNegVeria_AktProz,LTRIB[3]} };
EQUIVALENCE,{ {AktVeriVersuchAvl,STRIB[1]}, {AktVeriVersuchPRegel,STRIB[2]},
  {AktVeriVERgebnis,STRIB[3]} };
EQUIVALENCE,{ {LokGvon10i2j32s0u0zu10i2j36s0u0,XX[3]} };
;Anwendungsvoraussetzungsliste Avl233:
ARRAY,233,6,{30,31,32,33,34,35};
;Initialisierung globaler Variablen:
INTLC,{ {AktAnzNegVeria10i2j33s0u0,0} };
INTLC,{ {bErfuellungAv30,0}, {bErfuellungAv31,0}, {bErfuellungAv32,0},
  {bErfuellungAv33,0} };
INTLC,{ {bErfuellungAv34,0}, {bErfuellungAv35,0} };
INTLC,{ {KA_Av30,"K"}, {KA_Av31,"K"}, {KA_Av32,"K"}, {KA_Av33,"K"},
  {KA_Av34,"K"}, {KA_Av35,"U"} };
INTLC,{ {Knv10i2j33s0u0,4} };
;Simulationszeitraum: tp6:= 182160 bis tm1:= 526320
INITIALIZE,182160,526320,YES,,NO;
NET;
FIN;

```

³⁵⁶ Siehe dazu die entsprechenden einleitenden Hinweise im Anhang A-3.2.

A-3.4 AweSim SUMMARY-Report des Szenarios TP_VERI1

Aus Platzgründen wird dieser SUMMARY-Report hier (im Gegensatz zum Original) mit verkleinertem Schriftgrad wiedergegeben.

** AweSim SUMMARY REPORT **
Sat Sep 26 22:37:23 2009

Simulation Project : MCIPEC Demobeispiel
Modeler : Dr. Markus Puetz
Date : 17.09.2009
Scenario : TP_VERI1

Run number 1 of 1
Current simulation time : 526170.000000
Statistics cleared at time : 20880.000000

** FILE STATISTICS REPORT for scenario TP_VERI1 **

File Number	Label or Input Location	Average Length	Standard Deviation	Maximum Length	Current Length	Average Wait Time
70	RES. SINST7	0.000	0.000	0	0	0.000
80	RES. SINST8	0.000	0.000	0	0	0.000
90	RES. SINST9	0.000	0.000	0	0	0.000
140	RES. SINST14	0.000	0.000	1	0	0.000
0	Event Calendar	2.000	0.000	2	1	72184.286

** ACTIVITY STATISTICS REPORT for scenario TP_VERI1 **

Activity Number	Label or Input Location	Average Utilization	Standard Deviation	Entity Count	Maximum Utilization
1	Line 23	0.000	0.000	0	0
2	Line 25	0.000	0.000	1	1

Activity Number	Current Utilization
1	0
2	0

** RESOURCE STATISTICSREPORT for scenario TP_VERI1 **

Resource Number	Resource Label	Average Util.	Standard Deviation	Current Util.	Maximum Util.
7	SINST7	0.000	0.000	0	0
8	SINST8	0.000	0.000	0	0
9	SINST9	0.000	0.000	0	0
14	SINST14	0.000	0.008	0	1

Resource Number	Current Capacity	Average Available	Current Available	Minimum Available	Maximum Available
7	1	1.000	1	1	1
8	1	1.000	1	1	1
9	1	1.000	1	1	1
14	1	1.000	1	0	1

A-3.5 Daten der Ausgabedatei VEAvl233.dat bezüglich des mit dem Szenario TP_VERI1 durchgeführten Simulationsexperimentes

Diese Ausgabedatei besteht aus dem folgenden mit Anführungszeichen wiedergegebenen Datensatz:

```
" 526170.00    AVL233    Pr14    PartialFALSE    1    1    1    0    1    0    0"
```

A-3.6 AweSim SUMMARY-Report des Szenarios TP_VERI2

Wie bereits in Anhang A-3.4 gehandhabt, wird auch dieser SUMMARY-Report nachfolgend (im Gegensatz zum Original) mit verkleinertem Schriftgrad wiedergegeben.

```
** AweSim SUMMARY REPORT **
Sat Sep 26 22:38:04 2009
```

```
Simulation Project : MCIPEC Demobeispiel
Modeler : Dr. Markus Puetz
Date : 17.09.2009
Scenario : TP_VERI2
```

```
Run number 1 of 1
Current simulation time      : 526170.000000
Statistics cleared at time  : 182160.000000
```

```
** FILE STATISTICS REPORT for scenario TP_VERI2 **
```

File Number	Label or Input Location	Average Length	Standard Deviation	Maximum Length	Current Length	Average Wait Time
70	RES. SINST7	0.000	0.000	0	0	0.000
80	RES. SINST8	0.000	0.000	0	0	0.000
90	RES. SINST9	0.000	0.000	0	0	0.000
140	RES. SINST14	0.000	0.000	1	0	0.000
0	Event Calendar	2.000	0.000	2	1	49144.286

```
** ACTIVITY STATISTICS REPORT for scenario TP_VERI2 **
```

Activity Number	Label or Input Location	Average Utilization	Standard Deviation	Entity Count	Maximum Utilization
1	Line 23	0.000	0.000	1	1
3	Line 25	0.000	0.000	0	0

Activity Number	Current Utilization
1	0
3	0

```
** RESOURCE STATISTICS REPORT for scenario TP_VERI2 **
```

Resource Number	Resource Label	Average Util.	Standard Deviation	Current Util.	Maximum Util.
7	SINST7	0.000	0.000	0	0
8	SINST8	0.000	0.000	0	0

9	SINST9	0.000	0.000	0	0
14	SINST14	0.000	0.009	0	1
Resource Number	Current Capacity	Average Available	Current Available	Minimum Available	Maximum Available
7	1	1.000	1	1	1
8	1	1.000	1	1	1
9	1	1.000	1	1	1
14	1	1.000	1	0	1

A-3.7 Daten der Ausgabedatei VEAvl233.dat bezüglich des mit dem Szenario TP_VERI2 durchgeführten Simulationsexperimentes

Diese Ausgabedatei besteht aus dem folgenden mit Anführungszeichen wiedergegebenen Datensatz:

```
„ 526170.00    AVL233      Pr14  PartialTRUE    4  1  1  1  1  1  0“
```

A-3.8 Quellcode des benutzerspezifischen C-Programms UI_VEVA

```
#include "vslam.h"

#define      AktVeriVAvlNr          LTRIB[1]
#define      AnzAvAktAvl           LTRIB[2]
#define      AktAnzNegVeria_AktProz LTRIB[3]
#define      AktVeriVERgebnis      STRIB[3]

/* Zusaetzliche Ersatztexte fuer Validitaetspruefungen (VP) */
#define      AnzProjektziele       LL[36]
#define      AktValiP_PB_BppzlARRAY LTRIB[5]
#define      AktValiPERgebnis      STRIB[6]

void SWFUNC EVENT(int iCode, ENTITY * peCur)
{
    int Pruefschritt, AvlSpalte, VarSpalte;
    int AnzKAv, AnzUAv, AnzAvE, AnzAvNE;
    int AnzKAvE, AnzKAvNE, AnzUAvE, AnzUAvNE;

    /* Zusaetzliche Variablenvereinbarungen fuer VP */
    int BppzlSpalte, AnzPbE, AnzPbNE, CheckZielwertArt, RefArrayNr;
    double Checkwert;

    Pruefschritt = 0;
    AvlSpalte = 1;
    AnzKAv = 0;
    AnzUAv = 0;
    AnzAvE = 0;
    AnzAvNE = 0;
    AnzKAvE = 0;
    AnzKAvNE = 0;
    AnzUAvE = 0;
    AnzUAvNE = 0;

    /* Zusaetzliche Variablenwertzuweisungen fuer VP */
    BppzlSpalte = 1;
    AnzPbE = 0;
    AnzPbNE = 0;

    switch (iCode)
    {
        case 1: /* Ankunft am EVENT-Knoten mit der Ereignis-Nr. 1 */
            /* Programmanweisungen für einen Verifikationsversuch */

            while (Pruefschritt < peCur->AnzAvAktAvl)
            {
                VarSpalte = GETARY(peCur->AktVeriVAvlNr,AvlSpalte);

                if (LL[VarSpalte] == 1) /* Aeussere if-else-Anweisung */
                {
                    ++ AnzAvE;

                    if (strcmp(SZ[VarSpalte],"K") == 0)
                    {
                        ++ AnzKAv;
                        ++ AnzKAvE;
                    }
                    else
                    {
                        ++ AnzUAv;
                        ++ AnzUAvE;
                    }
                }
            }
        }
    }
}
```

```

    }
}
else /* else-Zweig der aeusseren if-else-Anweisung */
{
    ++ AnzAvNE;

    if (strcmp(SZ[VarSpalte], "K") == 0)
    {
        ++ AnzKAv;
        ++ AnzKAvNE;
    }
    else
    {
        ++ AnzUAv;
        ++ AnzUAvNE;
    }
}

++ Pruefschritt;
++ AvlSpalte;
}

/* Generierung des Ergebnisses des Verifikationsversuchs */
if (AnzAvE == peCur->AnzAvAktAvl)

    peCur->AktVeriVERgebnis = "TotalTRUE";

else if (AnzAvNE == peCur->AnzAvAktAvl)
{
    peCur->AktVeriVERgebnis = "TotalFALSE";
    ++ peCur->AktAnzNegVeria_AktProz;
}
else if (AnzKAvE == AnzKAv)

    peCur->AktVeriVERgebnis = "PartialTRUE";

else
{
    peCur->AktVeriVERgebnis = "PartialFALSE";
    ++ peCur->AktAnzNegVeria_AktProz;
}

break;

case 2: /* Ankunft am EVENT-Knoten mit der Ereignis-Nr. 2 */
/* Programmanweisungen für eine Validitätspruefung */

RefArrayNr = peCur->AktValiP_PB_BppzlARRAY;

while (Pruefschritt < AnzProjektziele)
{
    Checkwert = GETARY(RefArrayNr + 3, BppzlSpalte);
    CheckZielwertArt = GETARY(RefArrayNr + 1, BppzlSpalte);

    if (CheckZielwertArt == 0)
    {
        if (Checkwert == 0)
            ++ AnzPbE;
        else
            ++ AnzPbNE;
    }
    else if (CheckZielwertArt == 1)
    {

```

```
        if (Checkwert <= 0)
            ++ AnzPbE;
        else
            ++ AnzPbNE;
    }
    else
    {
        if (Checkwert >= 0)
            ++ AnzPbE;
        else
            ++ AnzPbNE;
    }
    ++ Pruefschritt;
    ++ BppzlSpalte;
}

/* Generierung des Ergebnisses der Validitaetspruefung */
if (AnzPbE == AnzProjektziele)

    peCur->AktValiPErgebnis = "TRUE";

else if (AnzPbNE == AnzProjektziele)

    peCur->AktValiPErgebnis = "TotalFALSE";

else

    peCur->AktValiPErgebnis = "PartialFALSE";

break;
default:
    su_error(1,"Unknown event code");
break;
}
}
```

A-3.9 Steuerungsanweisungen der Steuerungsdatei CVAL_TP7

Die nachfolgend aufgelisteten Steuerungsanweisungen der Steuerungsdatei CVAL_TP7 sind wiederum in der Weise strukturiert, die bereits für die Auflistung der Steuerungsanweisungen der Steuerungsdateien CVER_TP2 und CVER_TP6 zugrunde gelegt wurde³⁵⁷.

Die Steuerungsdatei CVAL_TP7 enthält folgende Steuerungsanweisungen:

**GEN,"Dr. Markus Puetz","MCIPEC Demobeispiel",17.09.2009,1,YES,YES;
LIMITS,5,40,36,,5,6,100;
;Sprachliche Bezeichner fuer bereits im Szenarios TP_VERI2 genutzte globale Variablen:**
EQUIVALENCE,{ {AktAnzNegVeria10i2j33s0u0,LL[1]} };
EQUIVALENCE,{ {bErfuellungAv30,LL[30]}, {bErfuellungAv31,LL[31]},
 {bErfuellungAv32,LL[32]} };
EQUIVALENCE,{ {bErfuellungAv33,LL[33]}, {bErfuellungAv34,LL[34]},
 {bErfuellungAv35,LL[35]} };
EQUIVALENCE,{ {KA_Av30,SZ[30]}, {KA_Av31,SZ[31]}, {KA_Av32,SZ[32]},
 {KA_Av33,SZ[33]} };
EQUIVALENCE,{ {KA_Av34,SZ[34]}, {KA_Av35,SZ[35]} };
EQUIVALENCE,{ {Knv10i2j33s0u0,LL[2]} };
EQUIVALENCE,{ {LokGvon10i2j32s0u0zu10i2j33s0u0,XX[1]},
 {LokGvon10i2j32s0u0zu10i2j34s0u0,XX[2]} };
EQUIVALENCE,{ {LokGvon10i2j32s0u0zu10i2j36s0u0,XX[3]} };
;Zusaetzliche sprachliche Bezeichner fuer globale Variablen im Szenario TP_VALI:
EQUIVALENCE,{ {AnzProjektziele,LL[36]} };
EQUIVALENCE,{ {PB_Bppzl233,LL[37]}, {SB_Bppzl233,LL[38]},
 {SWA_Bppzl233,LL[39]} };
EQUIVALENCE,{ {LokGvon10i2j35s0u0zu10i2j37s0u0,XX[4]} };
EQUIVALENCE,{ {LokGvon10i2j35s0u0zu10i2j38s0u0,XX[5]} };
;Sprachliche Bezeichner fuer bereits im Szenario TP_VERI2 genutzte einheitenspezifische Variablen:
EQUIVALENCE,{ {AktVeriVAvlNr,LTRIB[1]}, {AnzAvAktAvl,LTRIB[2]},
 {AktAnzNegVeria_AktProz,LTRIB[3]} };
EQUIVALENCE,{ {Knv_AktProz,LTRIB[4]} };
EQUIVALENCE,{ {AktVeriVersuchAvl,STRIB[1]}, {AktVeriVersuchPRegel,STRIB[2]},
 {AktVeriVErgebnis,STRIB[3]} };
;Zusaetzliche sprachliche Bezeichner fuer einheitenspezifische Variablen im Szenario TP_VALI:
EQUIVALENCE,{ {AktValiP_Bppzl,STRIB[4]}, {AktValiP_PRegel,STRIB[5]} };
EQUIVALENCE,{ {AktValiP_PB_BppzlARRAY,LTRIB[5]} };
EQUIVALENCE,{ {AktValiP_Ergebnis,STRIB[6]} };
;Anwendungsvoraussetzungsliste Avl233:
ARRAY,233,6,{30,31,32,33,34,35};
;Zusaetzliche ARRAYS im Szenario TP_VALI:
;Planbeitraege zur Erreichung von Projektzielen bzgl. Bppzl233:
ARRAY,102330,4,{0.95,15,1,0};
;Zielwertarten der Beitrage zur Erreichung von Projektzielen bzgl. Bppzl233:

³⁵⁷ Siehe dazu die entsprechenden einleitenden Hinweise im Anhang A-3.2.

```

ARRAY,102331,4,{1,2,0,0};
;Simulierte Beitraege zur Erreichung von Projektzielen bzgl. Bppzl233:
ARRAY,102332,4,{0,0,0,0};
;Soll-Wird-Abweichungswerte zu den Positionen zur Erreichung von Projektzielen bzgl.
Bppzl233:
ARRAY,102333,4,{0,0,0,0};
;Initialisierung globaler Variablen, wie bereits im Szenario TP_VERI2 vereinbart:
INTLC,{{AktAnzNegVeria10i2j33s0u0,4}};
INTLC,{{bErfuellungAv30,0},{bErfuellungAv31,0},{bErfuellungAv32,0},
      {bErfuellungAv33,0}};
INTLC,{{bErfuellungAv34,0},{bErfuellungAv35,0}};
INTLC,{{KA_Av30,"K"},{KA_Av31,"K"},{KA_Av32,"K"},{KA_Av33,"K"},
      {KA_Av34,"K"},{KA_Av35,"U"}};
INTLC,{{Knv10i2j33s0u0,4}};
;Zusaetzliche Initialisierung globaler Variablen im Szenario TP_VALI:
INTLC,{{AnzProjektziele,4}};
INTLC,{{PB_Bppzl233,102330},{SB_Bppzl233,102332},{SWA_Bppzl233,102333}};
;Simulationszeitraum: tp7:= 222480 bis tm1:= 526320
INITIALIZE,222480,526320,YES,,NO;
NET;
FIN;

```

A-3.10 AweSim SUMMARY-Report des Szenarios TP_VALI

Wie bereits in Anhang A-3.4 und A-3.6 gehandhabt, wird auch dieser SUMMARY-Report nachfolgend (im Gegensatz zum Original) mit verkleinertem Schriftgrad wiedergegeben.

** AweSim SUMMARY REPORT **
Mon Nov 02 10:03:53 2009

Simulation Project : MCIPEC Demobeispiel
Modeler : Dr. Markus Puetz
Date : 17.09.2009
Scenario : TP_VALI

Run number 1 of 1
Current simulation time : 526260.000000
Statistics cleared at time : 222480.000000

** FILE STATISTICS REPORT for scenario TP_VALI **

File Number	Label or Input Location	Average Length	Standard Deviation	Maximum Length	Current Length	Average Wait Time
70	RES. SINST7	0.000	0.000	1	0	0.000
80	RES. SINST8	0.000	0.000	0	0	0.000
90	RES. SINST9	0.000	0.000	0	0	0.000
140	RES. SINST14	0.000	0.000	1	0	0.000
0	Event Calendar	2.000	0.000	2	1	37972.500

** ACTIVITY STATISTICS REPORT for scenario TP_VALI **

Activity Number	Label or Input Location	Average Utilization	Standard Deviation	Entity Count	Maximum Utilization
1	Line 40	0.000	0.000	1	1
2	Line 21	0.000	0.000	1	1
3	Line 42	0.000	0.000	0	0

Activity Number	Current Utilization
1	0
2	0
3	0

** RESOURCE STATISTICS REPORT for scenario TP_VALI **

Resource Number	Resource Label	Average Util.	Standard Deviation	Current Util.	Maximum Util.
7	SINST7	0.000	0.010	0	1
8	SINST8	0.000	0.010	0	1
9	SINST9	0.000	0.010	0	1
14	SINST14	0.000	0.017	0	1

Resource Number	Current Capacity	Average Available	Current Available	Minimum Available	Maximum Available
7	1	1.000	1	0	1
8	1	1.000	1	0	1
9	1	1.000	1	0	1
14	1	1.000	1	0	1

A-3.11 Daten der Ausgabedatei VEAvl233.dat bezüglich des mit dem Szenario TP_VALI durchgeführten Simulationsexperimentes

Diese Ausgabedatei besteht aus dem folgenden mit Anführungszeichen wiedergegebenen Datensatz:

```
„ 526170.00 AVL233 Pr14 PartialTRUE 4 1 1 1 1 1 0“
```

A-3.12 Daten der Ausgabedatei VAPTrueBppzl233.dat bezüglich des mit dem Szenario TP_VALI durchgeführten Simulationsexperimentes

Diese Ausgabedatei besteht aus dem folgenden mit Anführungszeichen wiedergegebenen Datensatz:

```
„ 526260.00 Bppzl233 Pr14 TRUE -0.01 3.00 0.0 0.0“
```

Metacontrolling im partizipativen
Entwicklungscontrolling
Ansätze zur formalsprachlichen Spezifikation und
simulationsgestützten Modellierung
Pütz, M.
2012, XXXIX, 500 S. 84 Abb., Softcover
ISBN: 978-3-8349-3133-7