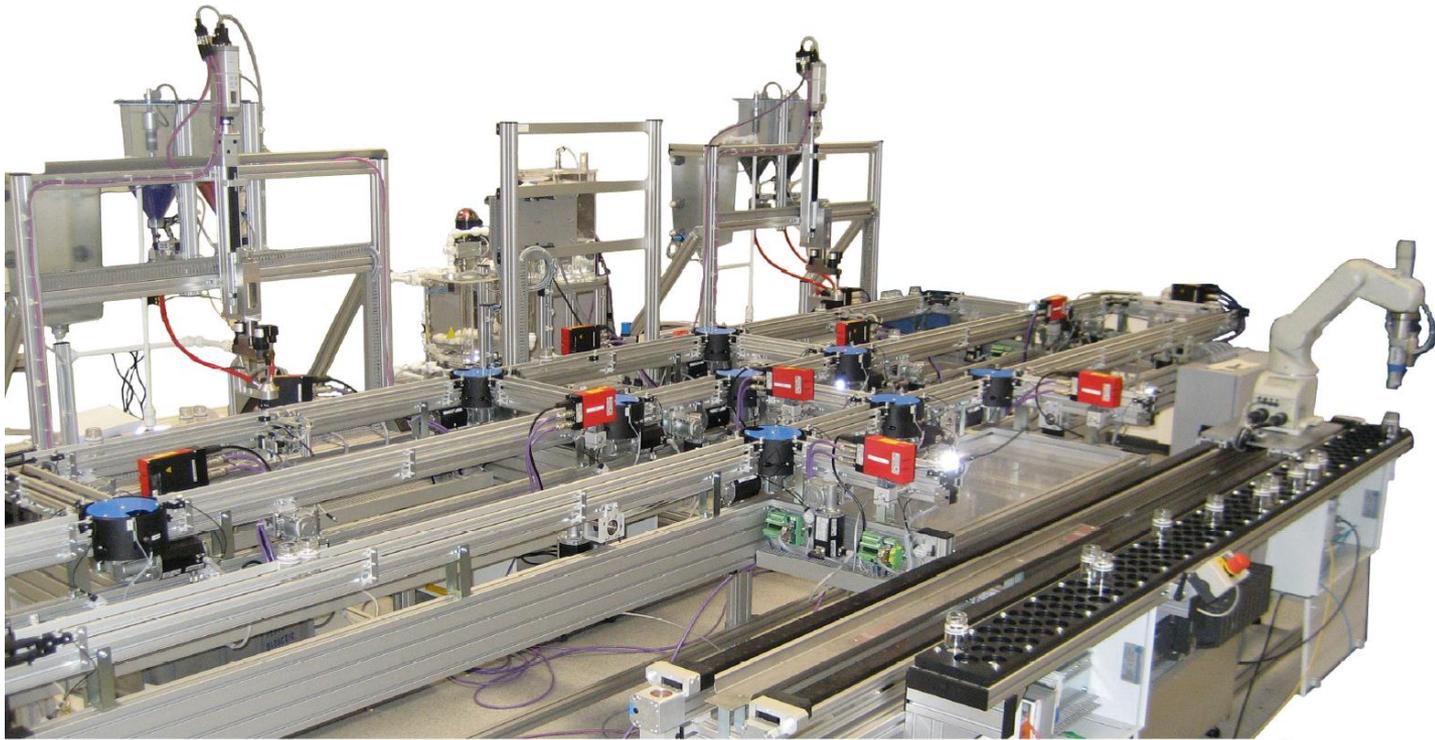


Entwurf, Implementierung und Evaluation eines modularen Agentensystems zur energieoptimierten Steuerung eines logistischen Automatisierungsprozesses



1. Motivation der Arbeit
2. Stand der Technik: Softwareagenten
 - a. Paradigma der Softwareagenten
 - b. Anwendungen von Softwareagenten
 - c. Entwurf von Agentensystemen
3. Entwurf des Agentensystems
4. Stand der Technik: Energetische Modellierung von Modulen
 - a. Bestehende Ansätze
 - b. Modellierungssprachen
5. Modellierung der Anlagenmodule
6. Implementierung des Agentensystems
7. Evaluation
8. Zusammenfassung und Ausblick

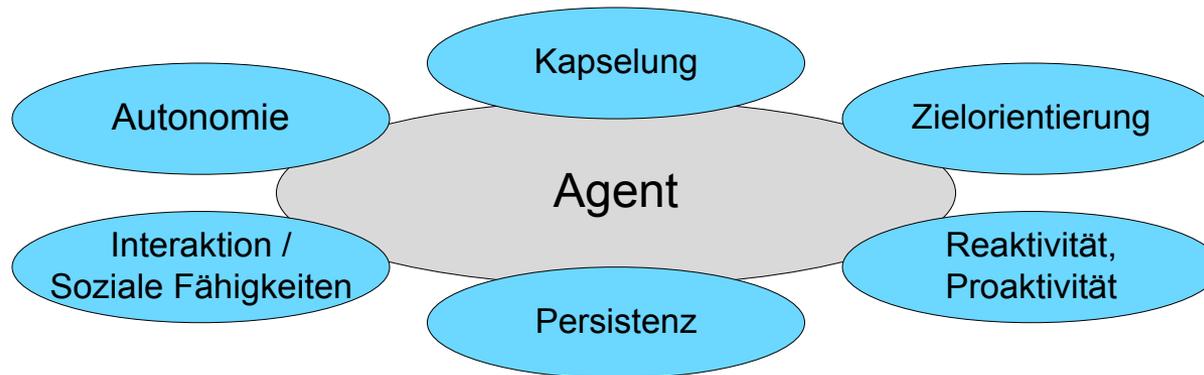
- Hoher Kostendruck durch Verteuerung der Energiepreise
 - Elektrische Energie verteuerte sich in Europa in 4 Jahren um 30% [Sei09]
- Durch Nutzung von regenerativen Energien:
 - Schwankungen der eingespeisten Leistung im Stromnetz [TBV09]
 - dynamische Veränderungen elektrischen Energiepreises [BNR06]
- Innovationen sind notwendig, Anlagen energetisch zu optimieren
 - auch vor dem Hintergrund eines dynamischen Energiepreises
- Aufdecken von Optimierungspotential anhand von Modellen
 - Modellierung von Struktur und Verhalten (energetische Aspekte)
 - Welche Form und Menge an Energie wird von welcher Komponenten für welche Teilfunktionen benötigt
- Softwareagenten erlauben flexible Auslegung der Steuerungssoftware
 - Softwareagenten können Optimierungsaufgaben dynamisch lösen
 - Grundlage dafür bildet ein Modell des zu optimierenden Systems



1. Motivation der Arbeit
2. **Stand der Technik: Softwareagenten**
 - a. Paradigma der Softwareagenten
 - b. Anwendungen von Softwareagenten
 - c. Entwurf von Agentensystemen
3. Entwurf des Agentensystems
4. Stand der Technik: Energetische Modellierung von Modulen
 - a. Bestehende Ansätze
 - b. Modellierungssprachen
5. Modellierung der Anlagenmodule
6. Implementierung des Agentensystems
7. Evaluation
8. Zusammenfassung und Ausblick



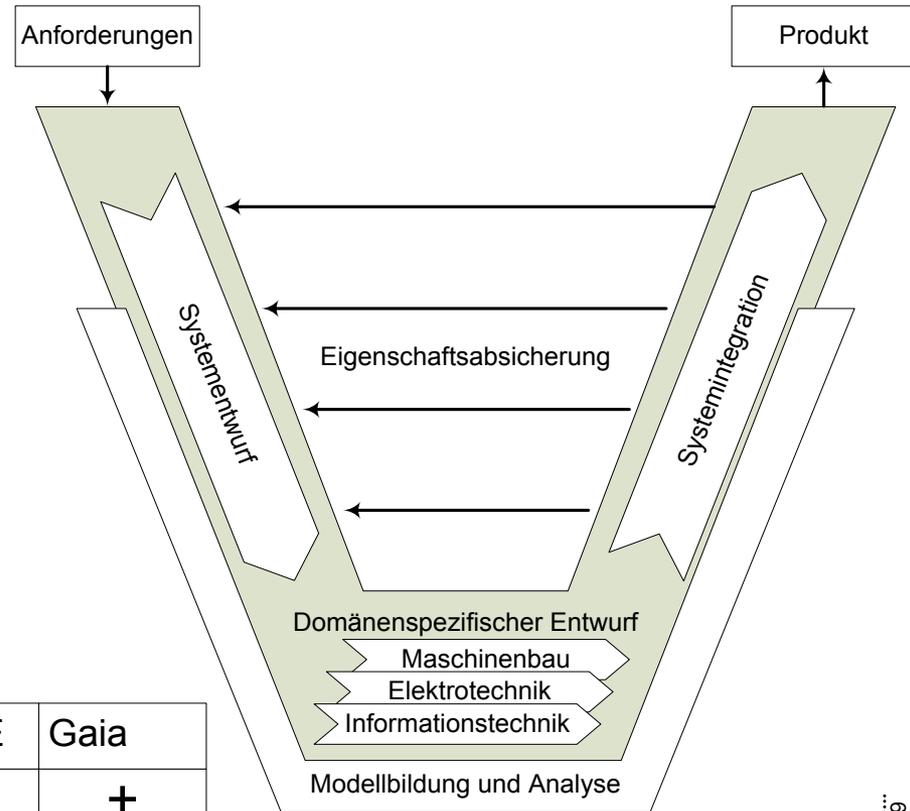
- Keine einheitliche Definition der Eigenschaften eines Softwareagenten
 - Schnittmenge von Eigenschaften, die in den verschiedenen Definitionen häufig genannt werden
- Verschiedene Architekturen für Softwareagenten
 - **Reaktive** Softwareagenten weisen ein regelbasiertes Verhalten auf und besitzen kein explizites Systemmodell [ScLo01]
 - **Deliberative** Agenten besitzen ein explizit formuliertes Modell ihrer Umwelt, auf dessen Grundlage sie ihre Ziele verfolgen [WoJe95]



Eigenschaften eines Agenten (vgl. [WoJe95], [HWH04])

- Wannagat et al.: Steigerung der Anlagenverfügbarkeit [WaVo08a]
 - Wissensbasis über mathematische Zusammenhänge von Sensorwerten
 - Überbrückung von Sensorausfällen durch virtuelle Sensoren
 - Oyarzabal et al.: Steuerung eines Niederspannungsnetzes [OJR+05]
 - Globaler Steuerungsagent überwacht das Niederspannungsnetz
 - Die Lasten der Verbraucher werden global zeitlich eingeplant
 - Generatoren bekommen Sollwerte vorgeschrieben
 - Heintel et al.: Bordnetzsteuerung eines Kfz durch Agenten [HWH04]
 - Ein Agent überwacht den Bordnetzstatus, gibt diese Informationen weiter
 - Einzelne Lasten werden durch Agenten gesteuert, melden ihren Energieverbrauch zusammen mit ihrer Priorität an
- Monetäre Verhandlung von Energie innerhalb des Agentensystems
- Agenten verfügen über energetisches Umweltmodell ihrer Anlagenmodule
 - Optimierung der Module auf deren Grundlage
 - Steuerungsagenten müssen Energie bei Versorgungsagent kaufen, Werkstückagent verlangt Voranschläge und vergibt Transportaufträge mit Zeit- und Kostenanforderungen

- Anforderungen an die Entwurfsmethode:
 - Umfang nicht bis hin zur Implementierung
 - Keine detaillierte Beschreibung der Protokolle
 - Keine Schritte der Anforderungs- oder Zielanalyse
 - Modellierung von Echtzeitanforderungen
 - Toolunterstützung



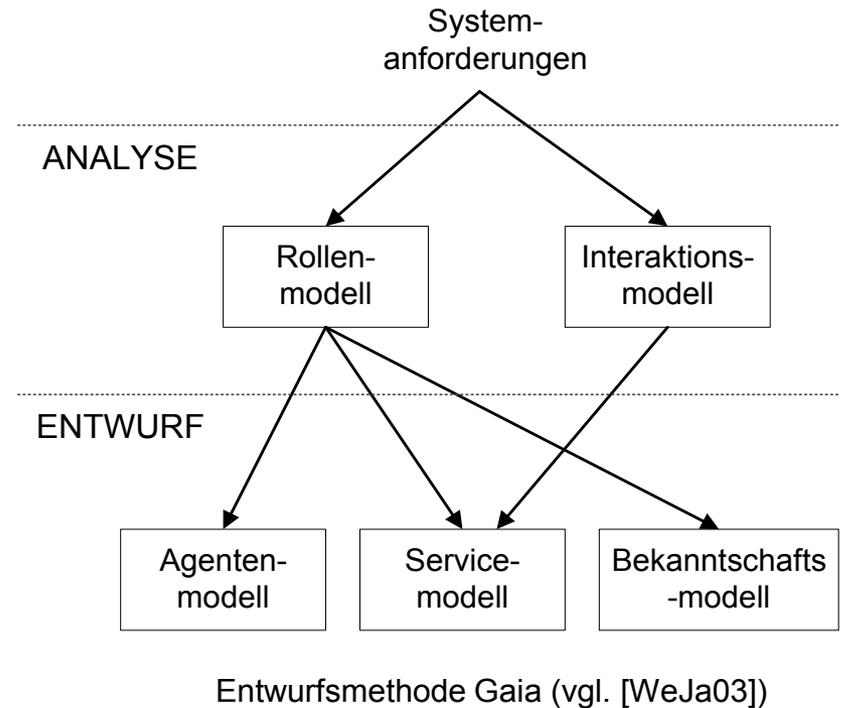
V-Modell nach VDI 2206

Ausführlicher Vergleich der Entwurfsmethoden: [MGW+07]
 Entwurfsmethode eMaSE: [BMG08]

Entwurfsmethode	PASSI	MaSE	eMaSE	Gaia
Passender Umfang	-	-	-	+
Keine detaillierte Protokollbeschreibung	-	-	-	+
Keine Anforderungs- / Zielanalyse	-	-	-	+
Echtzeitanforderungen	-	-	+	+
Toolunterstützung	+	+	+/-	-

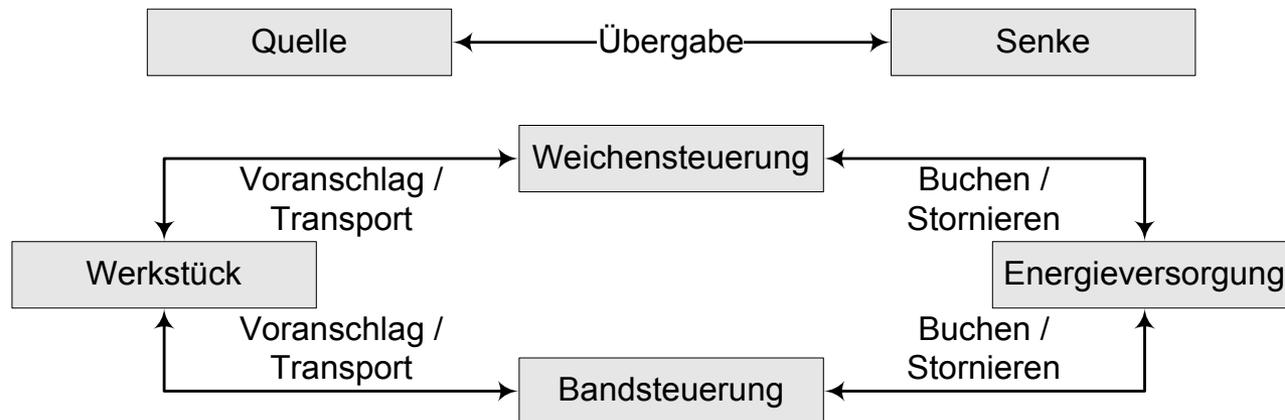
- Analysephase: iterative Verfeinerung von Rollenmodell und Interaktionsmodell
 - Definition der sozialen Rollen innerhalb des Agentensystems
 - Definition der Interaktionsprotokolle zwischen den einzelnen Rollen

- Entwurfsphase
 - Definition von Agententypen über Rollen, welche diese übernehmen
 - Definition der Dienste der einzelnen Agententypen
 - Identifizierung von Kommunikationsengpässen zwischen den Agententypen

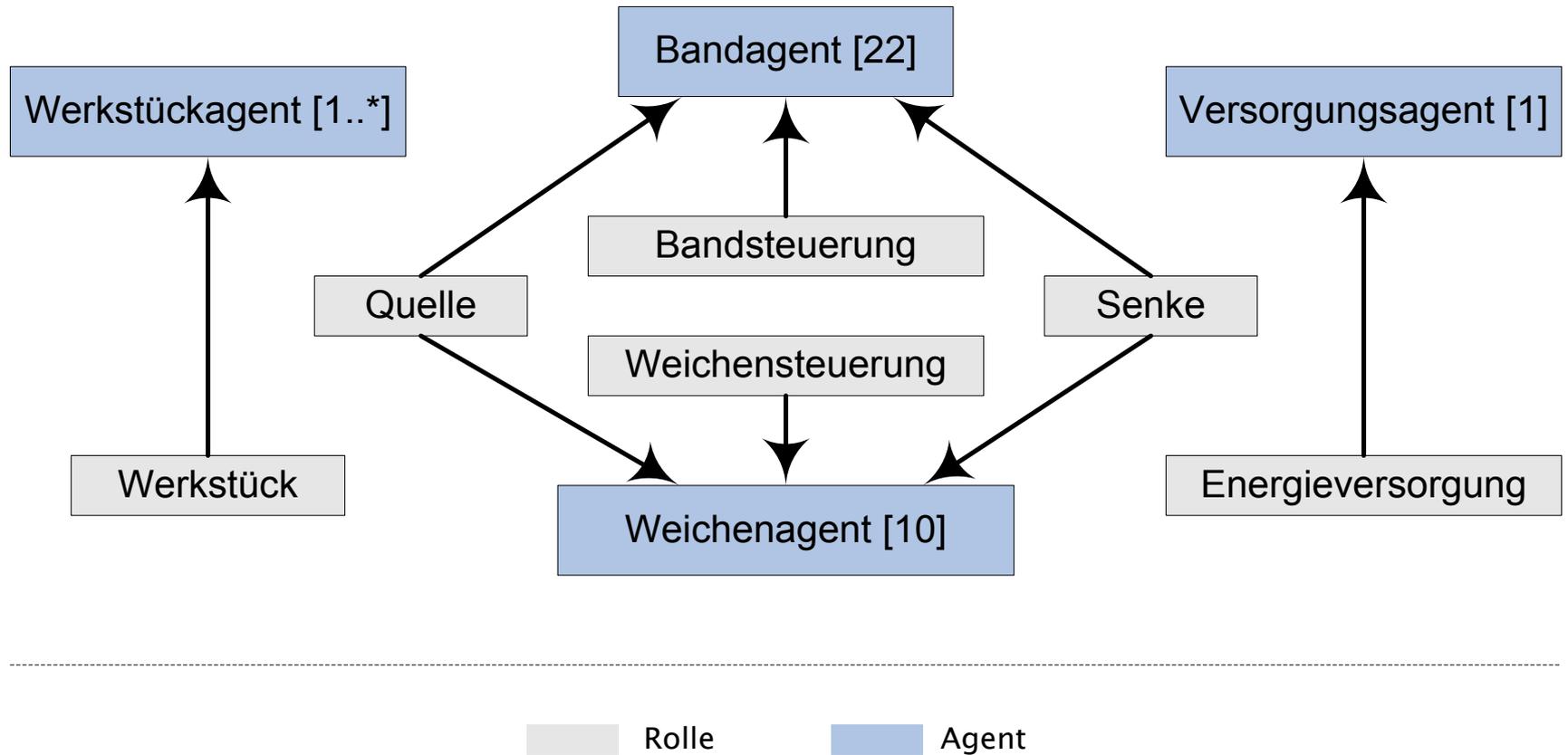


1. Motivation der Arbeit
2. Stand der Technik: Softwareagenten
 - a. Paradigma der Softwareagenten
 - b. Anwendungen von Softwareagenten
 - c. Entwurf von Agentensystemen
- 3. *Entwurf des Agentensystems***
4. Stand der Technik: Energetische Modellierung von Modulen
 - a. Bestehende Ansätze
 - b. Modellierungssprachen
5. Modellierung der Anlagenmodule
6. Implementierung des Agentensystems
7. Evaluation
8. Zusammenfassung und Ausblick

- Identifizierte Rollen im Agentensystem
 - Bandsteuerung / Weichensteuerung → Steuerung der Transportmodule
 - Energieversorgung → Berechnung des Energiepreises und Energiehandel
 - Werkstück → Steuerung des Prozesses, Zeit- / Kostenanforderungen
 - Quelle / Senke → Übergabe von Werkstücken zwischen zwei Transportmodulen
- Identifizierte Protokolle zwischen den Rollen
 - Übergabe → Regelung der Übergabe von Werkstücken
 - Voranschlag → Beziehen von Voranschlägen durch das Werkstück
 - Transport → Erteilung und Ausführung eines Transportauftrags
 - Buchen / Stornieren → Bezug von Leistung bei der Energieversorgung melden



- Im **Agentenmodell** werden die verschiedenen Agententypen und deren Anzahl im System definiert
 - Welcher Agententyp übernimmt welche Rollen

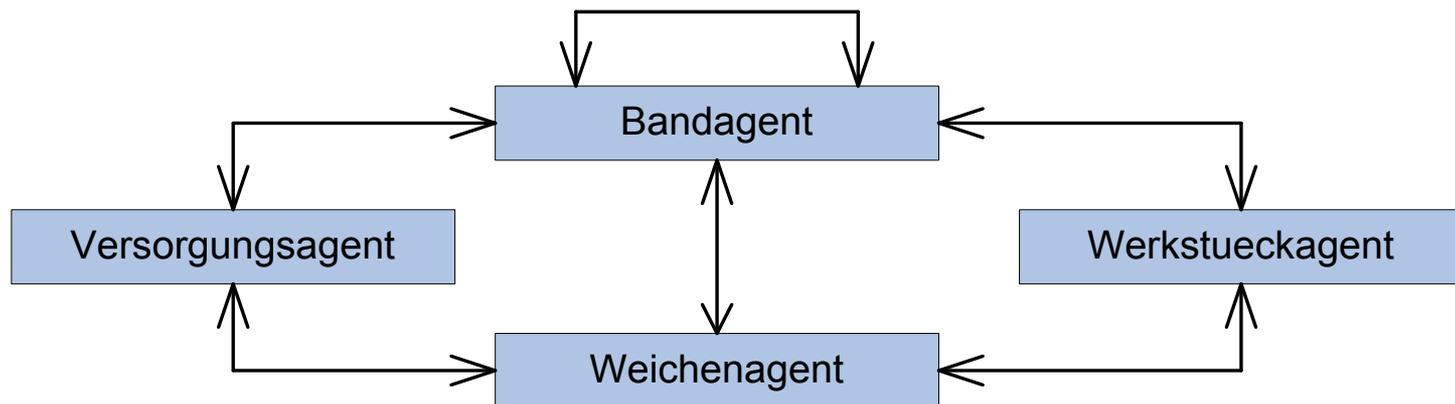


- Im **Servicemodell** werden die Dienste der einzelnen Agenten definiert
 - Abgeleitet aus Rollen- und Interaktionsmodell: Jedes Protokoll und jede Aktivität einer Rolle wird durch einen Dienst repräsentiert
 - Verfeinerung und Erweiterung durch weitere Dienste
- Beschreibung durch:
 - Ein- und Ausgabevariablen
 - Vor- und Nachbedingungen

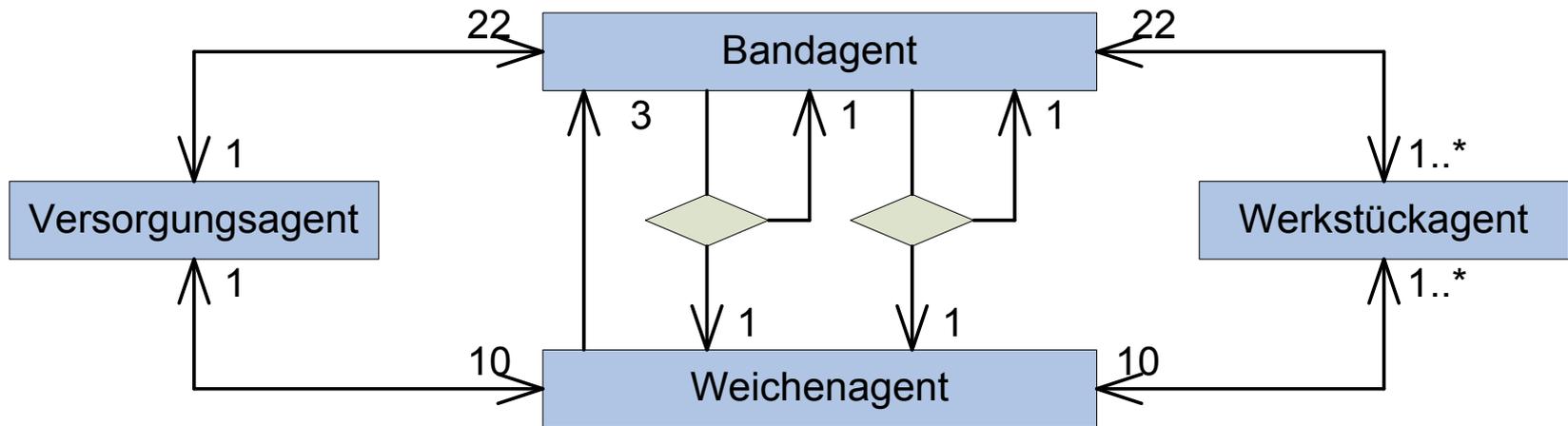
Servicemodell Bandagent (Ausschnitt):

Dienst	Input	Output	PreCon	PostCon
Aufnahmebereitschaft	<i>s abgabestatus, v_ist, a_ist</i>	<i>aufnahmestatus, v_soll, a_soll</i>	<i>s abgabestatus = bereit</i>	<i>aufnahmestatus = bereit</i>

- **Bekanntschftsmodell:** Identifizierung von Kommunikationsengpässen
 - Tatsächliche Kommunikationspfade zwischen den Instanzen der Agententypen und deren Anzahl wird nicht berücksichtigt



- **Bekanntschftsmodell:** Identifizierung von Kommunikationsengpässen
 - Tatsächliche Kommunikationspfade zwischen den Instanzen der Agententypen und deren Anzahl wird nicht berücksichtigt
- Erweiterung des Diagramms durch:
 - Darstellung der Kommunikationswege für einzelne Instanzen der Agententypen
 - Berücksichtigung der Anzahl der Instanzen der Agenten im System
 - Darstellung von instanzabhängigen Alternativen (Diamantnotation)



1. Motivation der Arbeit
2. Stand der Technik: Softwareagenten
 - a. Paradigma der Softwareagenten
 - b. Anwendungen von Softwareagenten
 - c. Entwurf von Agentensystemen
3. Entwurf des Agentensystems
4. **Stand der Technik: Energetische Modellierung von Modulen**
 - a. Bestehende Ansätze
 - b. Modellierungssprachen
5. Modellierung der Anlagenmodule
6. Implementierung des Agentensystems
7. Evaluation
8. Zusammenfassung und Ausblick

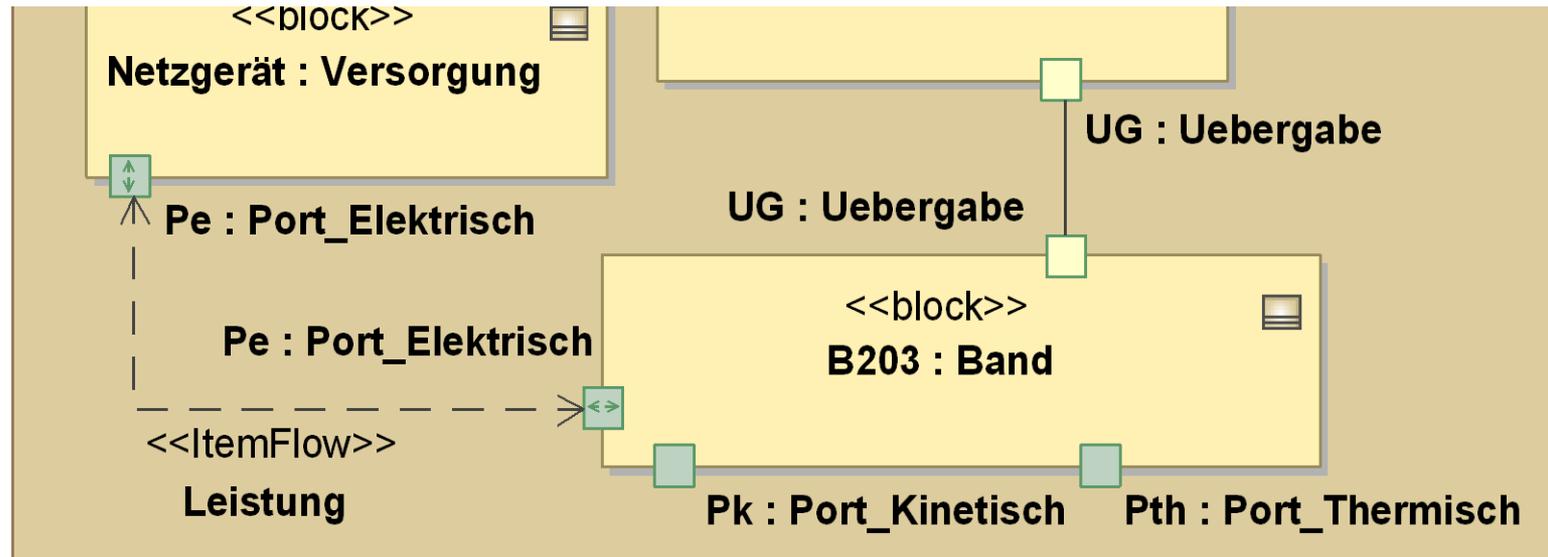
- Secchi et al.: Energetische Modellierung von Modulen mit UML-RT [SBF07]
 - Energieschnittstellen und –übertragung modelliert durch Ports und Konnektoren
 - Mathematische Zusammenhänge in Form von Constraints
 - Modellierung ausschließlich auf Klassenebene
- Schütz et al.: Energetische Modellierung von Modulen mit SysML [SVB+09]
 - Ergänzung um Modellierung auf Objektebene, mathematische Zusammenhänge in Constraint-Blöcken (Parameterdiagramm)
 - Erweiterung der Schnittstellenmodellierung durch SysML-Stereotypen
- Schütz et al.: Modellierung automatisierungstechnischer Module [ScWa09]
 - Beschreibung des inneren Verhaltens von Modulen durch Zustandsdiagramme
 - Inneres Verhalten ermöglicht Modulen ihre Funktionen auszuführen

- Modellierung der energetischen Struktur der Module auf Objektebene
- parameterbasiertes energetisches Verhalten der Module (Übertragung / Wandlung von Energie)
- zustandsbasiertes inneres Verhalten der Module (energetisch bewertet)
- funktionales Verhalten (energetisch optimierte Funktionsausführung)
 - Modellierung einer energetischen Bilanzierung von Funktionen

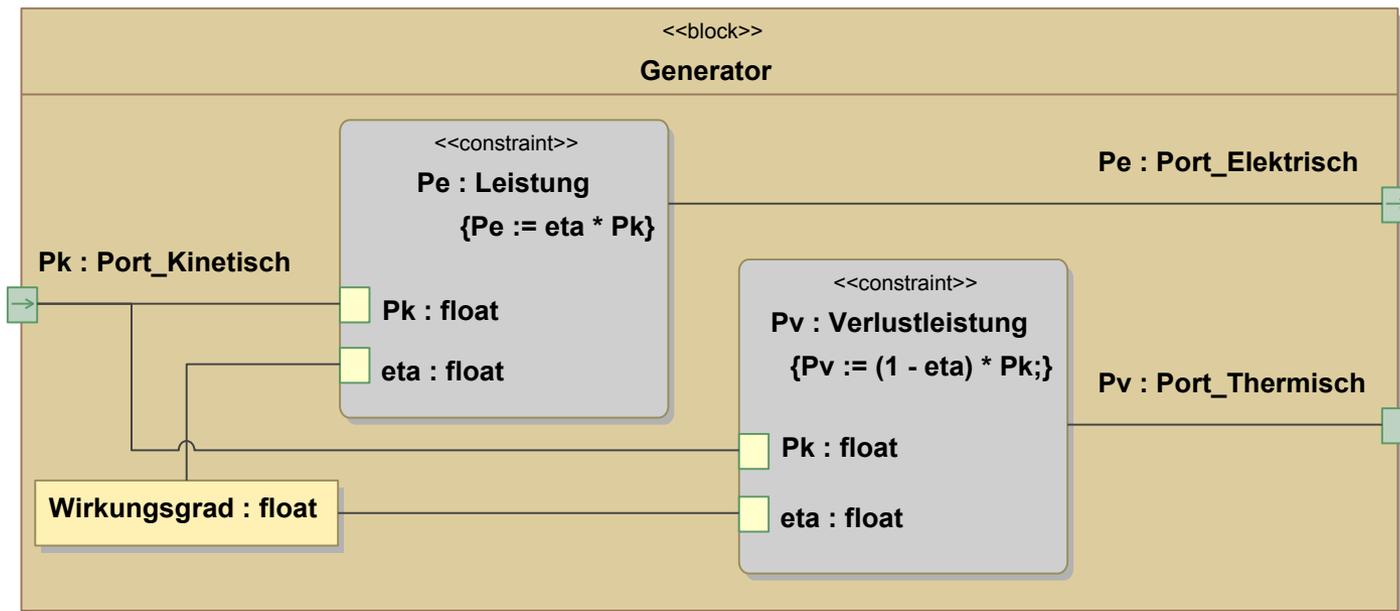
	UML 2.2	UML-RT	SysML
zustandsbasiertes Verhalten	Ja, Zustandsdiagramm	Ja, Zustandsdiagramm	Ja, Zustandsdiagramm
zeitliches Verhalten	Timing-Diagramm	Timing-Diagramm	/
Strukturmodellierung auf Objektebene	Kompositionsstrukturdiagramm	Capsule-Diagramm	Internes Blockdiagramm
Zusammenhänge von Parametern	Constraints	Constraints	Constraints-Blöcke, Parameterdiagramm
Modulschnittstellen, Verbindungen	Ports, Konnektoren	Ports, Konnektoren	Ports, Konnektoren
physikalische Schnittstellen	/	/	Flowports, Itemflows

1. Motivation der Arbeit
2. Stand der Technik: Softwareagenten
 - a. Paradigma der Softwareagenten
 - b. Anwendungen von Softwareagenten
 - c. Entwurf von Agentensystemen
3. Entwurf des Agentensystems
4. Stand der Technik: Energetische Modellierung von Modulen
 - a. Bestehende Ansätze
 - b. Modellierungssprachen
- 5. *Modellierung der Anlagenmodule***
6. Implementierung des Agentensystems
7. Evaluation
8. Zusammenfassung und Ausblick

- Modellierung der **Struktur** in internen Blockdiagrammen (Objektebene)
 - Module als Blöcke dargestellt
 - Datenschnittstellen und Übertragung modelliert durch Ports und Konnektoren
 - Energieschnittstellen und Übertragung modelliert durch Flowports und Itemflows (Abgrenzung gegenüber Datenschnittstellen)

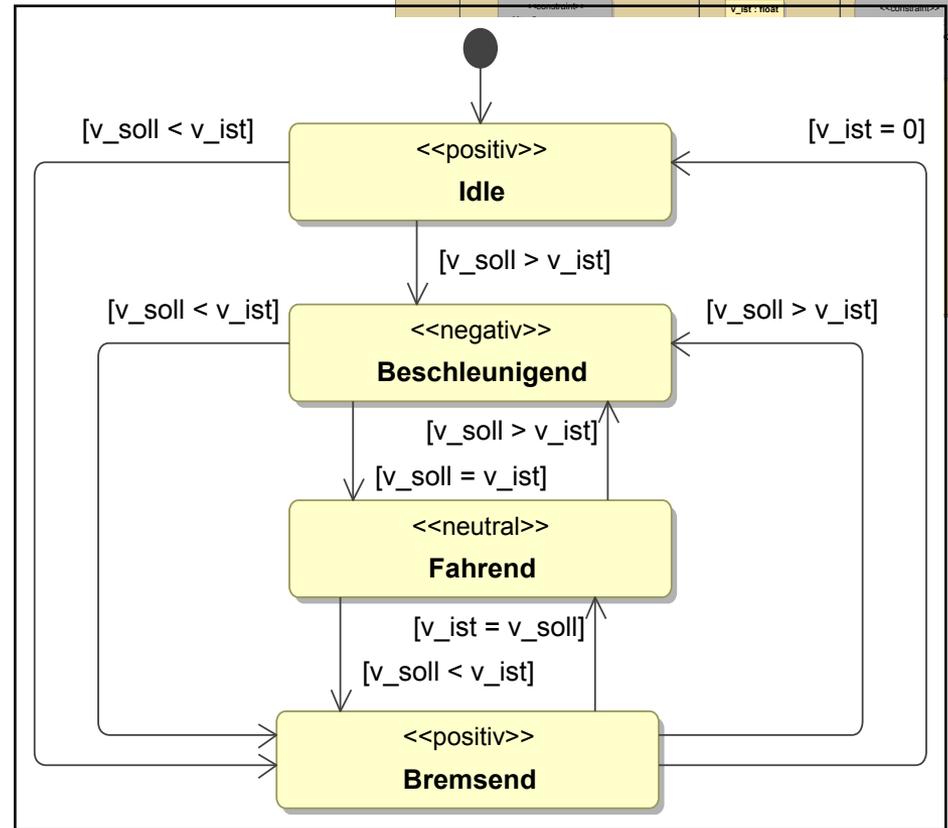
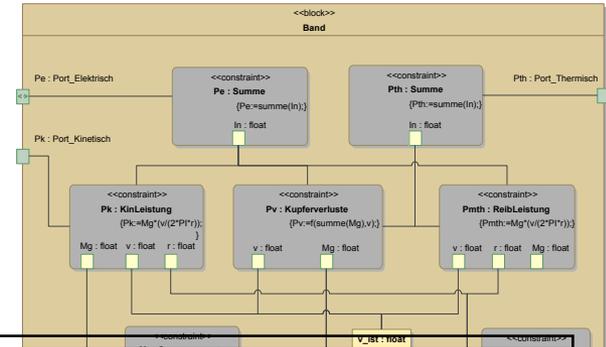


- Modellierung des **parameterbasierten Verhaltens** auf Objektebene im Parameterdiagramm
 - Parameter sind gegeben durch Attribute des Blocks
 - Mathematische Zusammenhänge formuliert durch Zusicherungen in Constraint-Blöcken
 - Verknüpfung der Parameter mit den Constraint-Blöcken durch Konnektoren
 - Verknüpfung der Energieschnittstellen mit den Constraints, welche die Leistung am jeweiligen Flowport berechnen



- Modellierung des **zustandsbasierten Verhaltens** der Module Transportband und Weiche durch Zustandsdiagramme
 - Ermöglicht Beschreibung eines Transports durch Zustandsfolge

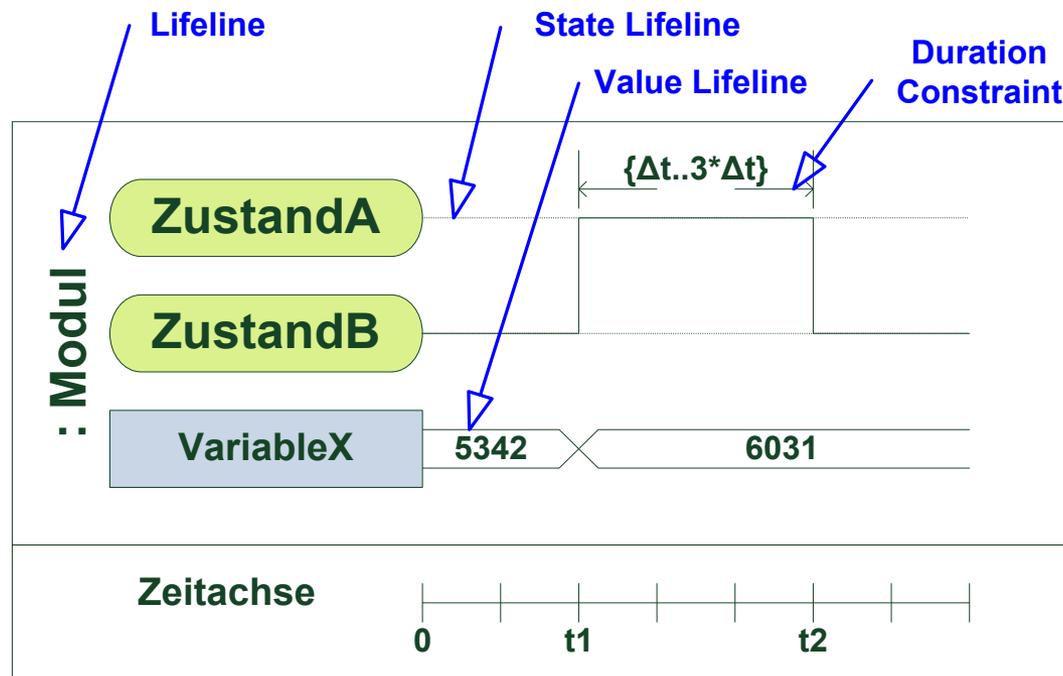
- Energetische Bewertung der Zustände auf Grundlage des Parameterdiagramms
- z.B. Kennzeichnung durch Stereotypen
- Vermeidung von Zuständen mit hohem Energieverbrauch bei der Funktionsausführung



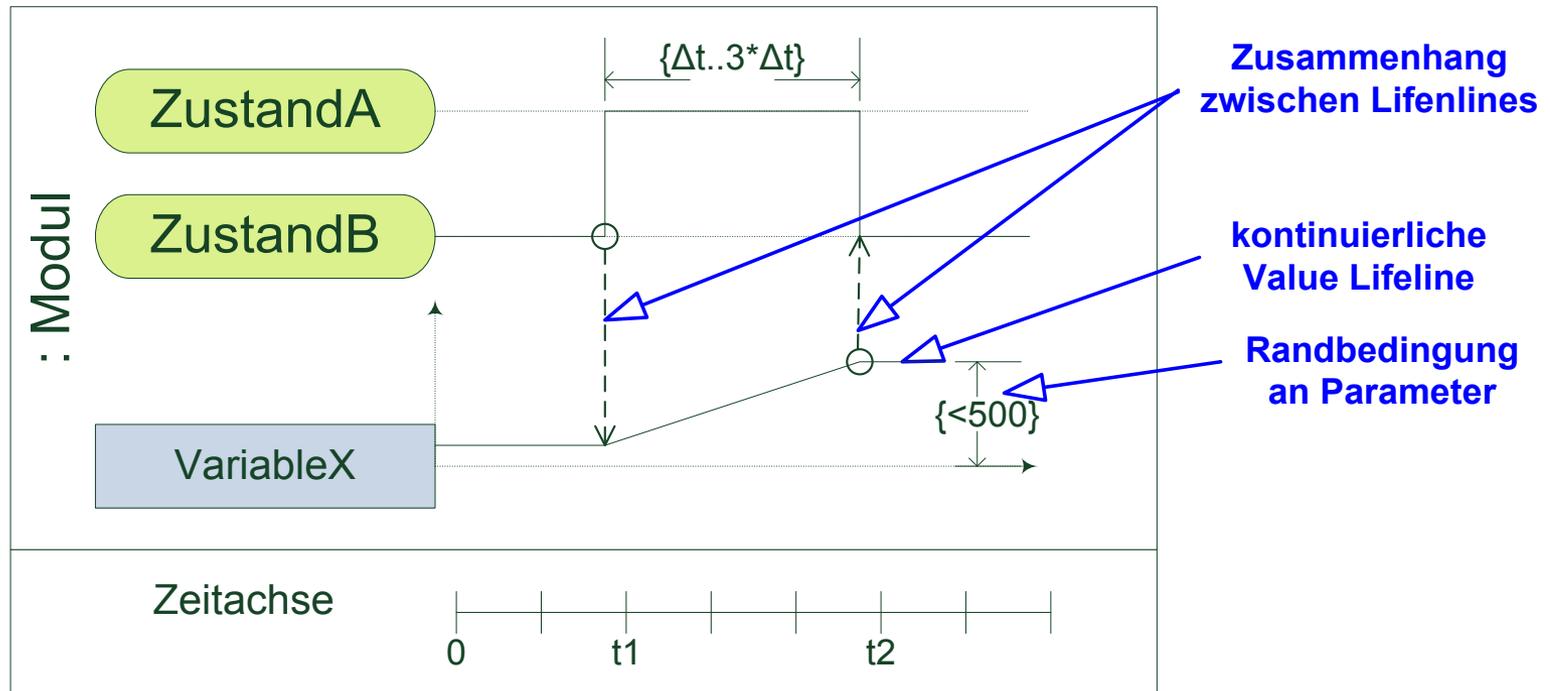
- Im **funktionalen Verhalten** wird die Ausführung der Funktionen der Module durch Zustandswechsel ihres inneren Verhaltens beschrieben
 - Funktionsausführung soll energetisch optimiert werden
- Transportband und Weiche müssen alle Zustände bei der Durchführung eines Transports durchlaufen
 - Energetische Optimierung durch Vermeidung von Zuständen mit hohem Energieverbrauch nicht möglich
- Optimierung besteht aus der Anpassung der Zeitintervalle in denen die einzelnen Zustände aktiv sind (wie lange wird beschleunigt, etc.)
 - Keine Möglichkeit zur Beschreibung mit den Diagrammen der SysML
- Beschreibung eines Verhaltensszenarios gegeben durch Zustandswechsel
 - Beschreibung über einer Zeitachse
 - Berücksichtigung der Wertverläufe der Parameter eines Moduls
 - Formulierung von Randbedingungen

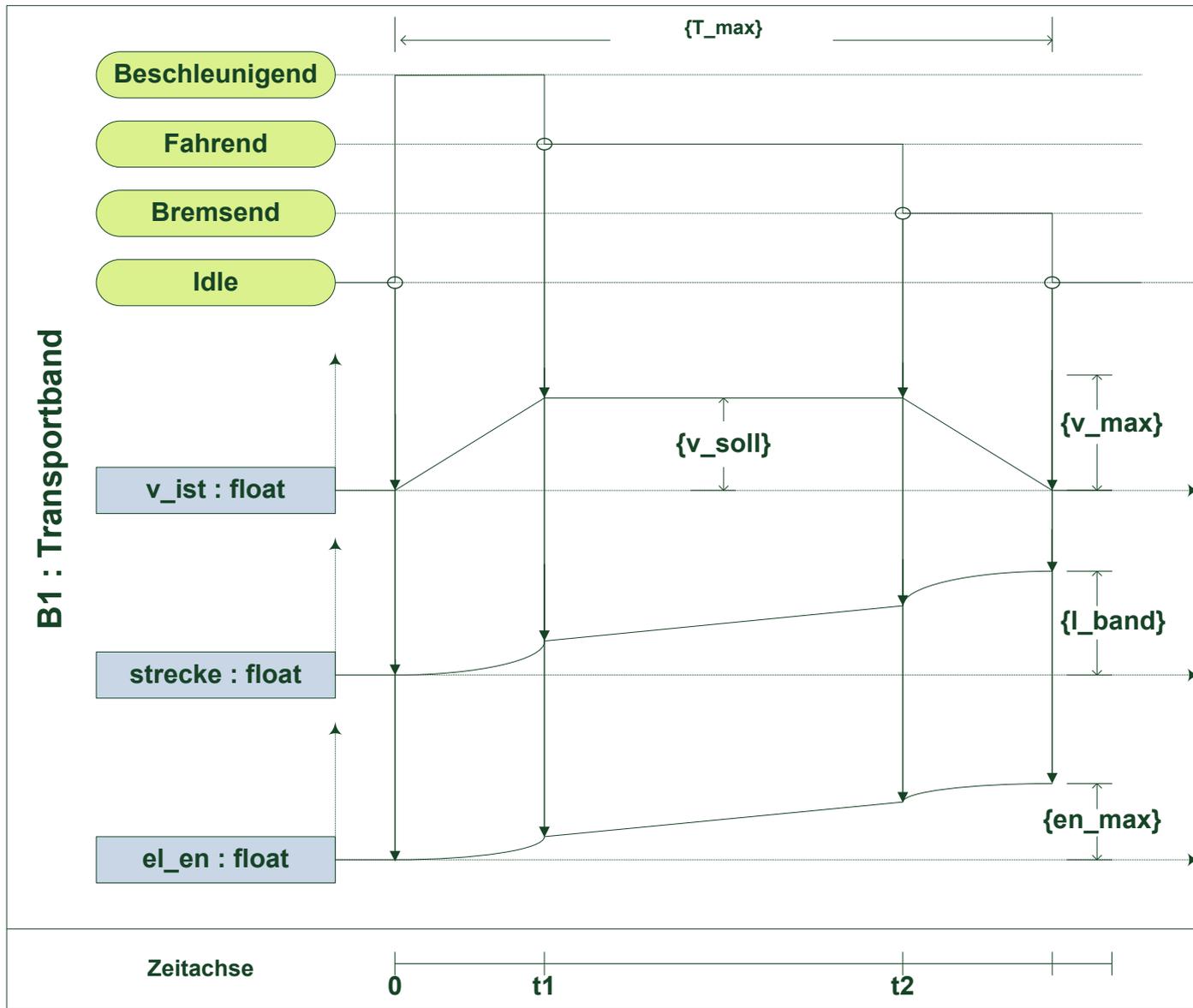


- Im Timing-Diagramm der UML können Verhaltensszenarien von Modulen über einer Zeitachse dargestellt werden
 - Darstellung durch Lifelines von Modulen und State Lifelines der Zustände
- Werteverläufe von Parametern eines Moduls in Form von Value Lifelines
 - Nur diskrete, keine kontinuierliche Darstellung von Werten
- Formulierung zeitlicher Randbedingungen durch Duration Constraints



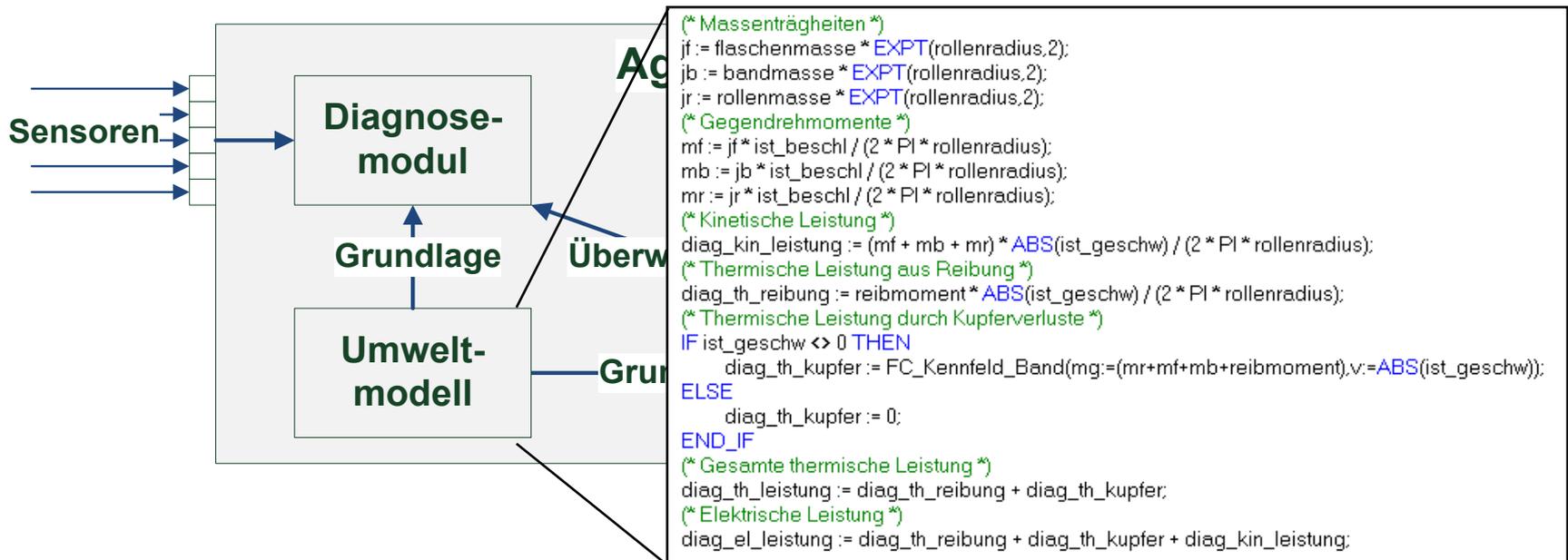
- Erweiterung der Notationen des Timing-Diagramms
 - Kontinuierliche Darstellung der Value Lifelines
 - Formulierung von Randbedingungen an Parameter des Moduls
 - Darstellung der Zusammenhänge zwischen Zustandswechseln und Werteverlauf von Parametern





1. Motivation der Arbeit
2. Stand der Technik: Softwareagenten
 - a. Paradigma der Softwareagenten
 - b. Anwendungen von Softwareagenten
 - c. Entwurf von Agentensystemen
3. Entwurf des Agentensystems
4. Stand der Technik: Energetische Modellierung von Modulen
 - a. Bestehende Ansätze
 - b. Modellierungssprachen
5. Modellierung der Anlagenmodule
- 6. *Implementierung des Agentensystems***
7. Evaluation
8. Zusammenfassung und Ausblick

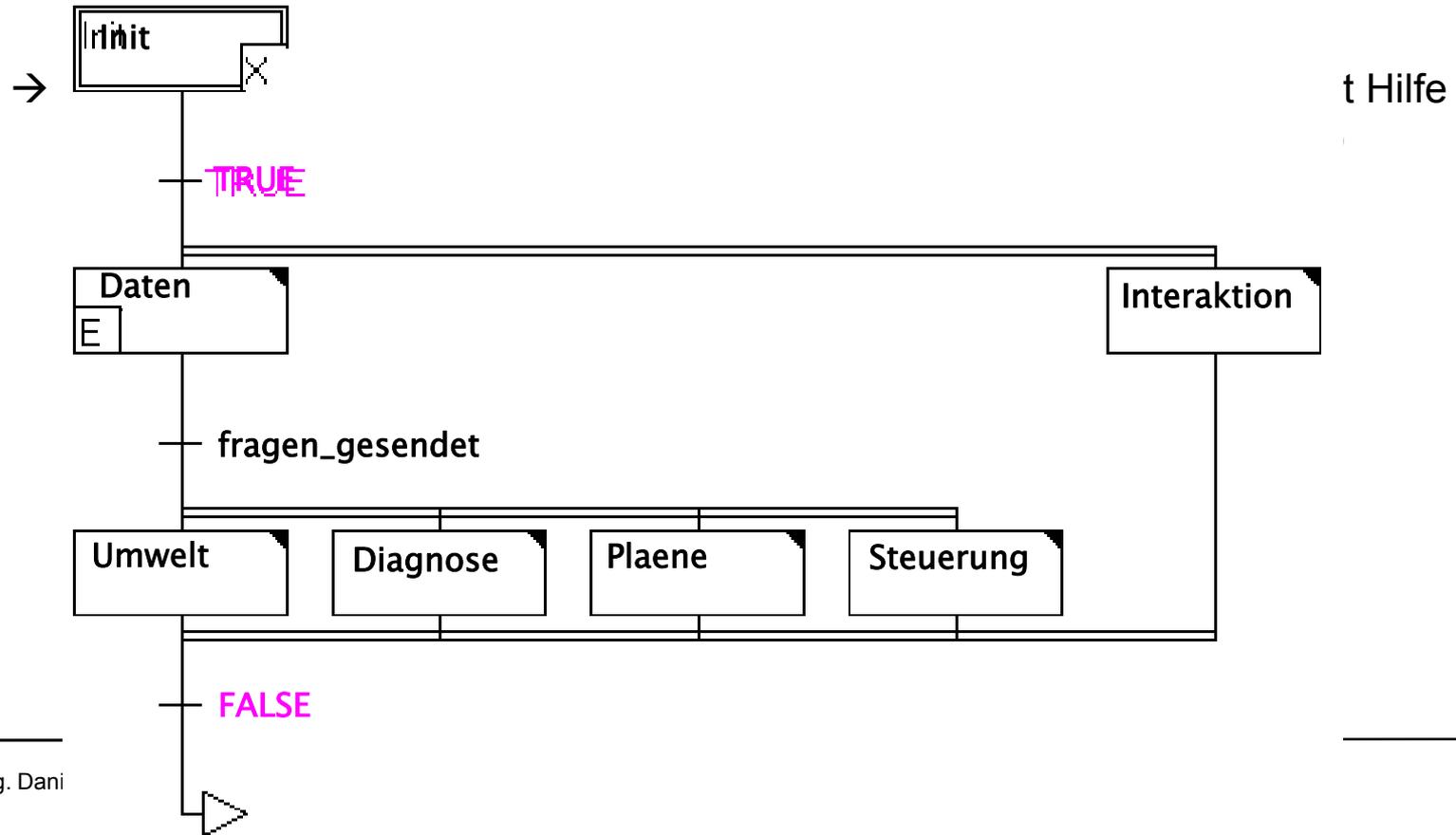
- Implementierung auf einem Laufzeitsystem nach IEC 61131
 - Agenten werden als Funktionsbaustein implementiert
- Implementierung von zusätzlichen Agenten gegenüber dem Entwurf (Anlehnung an Wannagat et al. [WaVo08a], [WaVo08b])
 - Kommunikationsagent → Weiterleitung von Nachrichten zwischen den Agenten (Mechanismus einer nachrichtenbasierten Kommunikation → FB_Mailbox)
 - Whiteboard → globale Verbreitung von Nachrichten (Zugriff für alle Agenten)
 - Systemagent → Informationen über technischen Aufbau der Anlage



- Umweltmodell als separaten Funktionsbaustein implementiert
 - verschiedene Aktivitäten des Rollenmodells als Aktionen implementiert
 - innerhalb eines Arrays für verschiedene Transportanforderungen bereits Parametersätze vordefiniert

→ Agenten besitzen hybride Architektur deliberativ-reaktiv

→ mit reaktivem Verhalten aufgrund von Regeln (Parametersätze) können die



1. Motivation der Arbeit
2. Stand der Technik: Softwareagenten
 - a. Paradigma der Softwareagenten
 - b. Anwendungen von Softwareagenten
 - c. Entwurf von Agentensystemen
3. Entwurf des Agentensystems
4. Stand der Technik: Energetische Modellierung von Modulen
 - a. Bestehende Ansätze
 - b. Modellierungssprachen
5. Modellierung der Anlagenmodule
6. Implementierung des Agentensystems
- 7. *Evaluation***
8. Zusammenfassung und Ausblick

- Flexible Auslegung der Steuerungssoftware
 - Veränderung / Erweiterung des Aufbaus der Anlage
- Energetische Optimierung der Funktionen der Module
 - Optimierung zur Erfüllung verschiedener Anforderungen zur Laufzeit
 - echtzeitfähig durch deliberativ-reaktive Mischarchitektur
- Energetisch optimierte Steuerung der Anlage
 - Produktion abhängig von Energiepreisen
 - Detaillierte Überwachung der Produktionskosten
 - Vermeidung von Energiespitzen innerhalb der Anlage
- Entwurf des Agentensystems mit der Methode Gaia
 - Implizite Formulierung der Echtzeiteigenschaften
 - Erweiterung der Modelle um wichtige Aspekte
- Energetische Modellierung der Module
 - Beschreibung der energetischen Aspekte eines Moduls in einem Modell
 - Erweiterung des Timing-Diagramms zur Beschreibung der Funktionsoptimierung

1. Motivation der Arbeit
2. Stand der Technik: Softwareagenten
 - a. Paradigma der Softwareagenten
 - b. Anwendungen von Softwareagenten
 - c. Entwurf von Agentensystemen
3. Entwurf des Agentensystems
4. Stand der Technik: Energetische Modellierung von Modulen
 - a. Bestehende Ansätze
 - b. Modellierungssprachen
5. Modellierung der Anlagenmodule
6. Implementierung des Agentensystems
7. Evaluation
- 8. Zusammenfassung und Ausblick**

- Entwurf und Implementierung eines Agentensystems zur energieoptimierten Steuerung eines logistischen Anlagenprozesses
 - Realisierung verschiedener positiver Verhaltensweisen der Anlage
 - Hybride Agentenarchitektur zur Einhaltung von Echtzeitanforderungen
- Energetische Modellierung von Anlagenmodulen in Diagrammen der Modellierungssprachen SysML / UML
 - Notation des Timing-Diagramms der UML wurde um Darstellungen erweitert
- Übertragung des Ansatzes auf die Beispiele anderer Anlagen
 - Verschiedene Energieformen, begrenzte Ressourcen, Verfahrenstechnik
- Ausbau der Methode Gaia im Bereich der Automatisierungstechnik
- Automatische Codegenerierung der Umweltmodelle aus der Modellierung
- Verwendung des Timing-Sequenz Diagramms der UML-PA [Kat09]
 - Stärkere Bindung der Lifelines an das Typenmodell



Vielen Dank
für Ihre Aufmerksamkeit

Role Schema: BANDSTEUERUNG (BS)**Description: ...****Protocols and Activities:**Buchen, Stornieren, Voranschlag, Transport, Bdiagnose, Bsetoptimum**Permission:**

reads $a_ist, v_ist, pos_a, pos_b, btech_param$
 supplied $en_preis, en_trend, anfrage, auftrag$

changes a_soll, v_soll

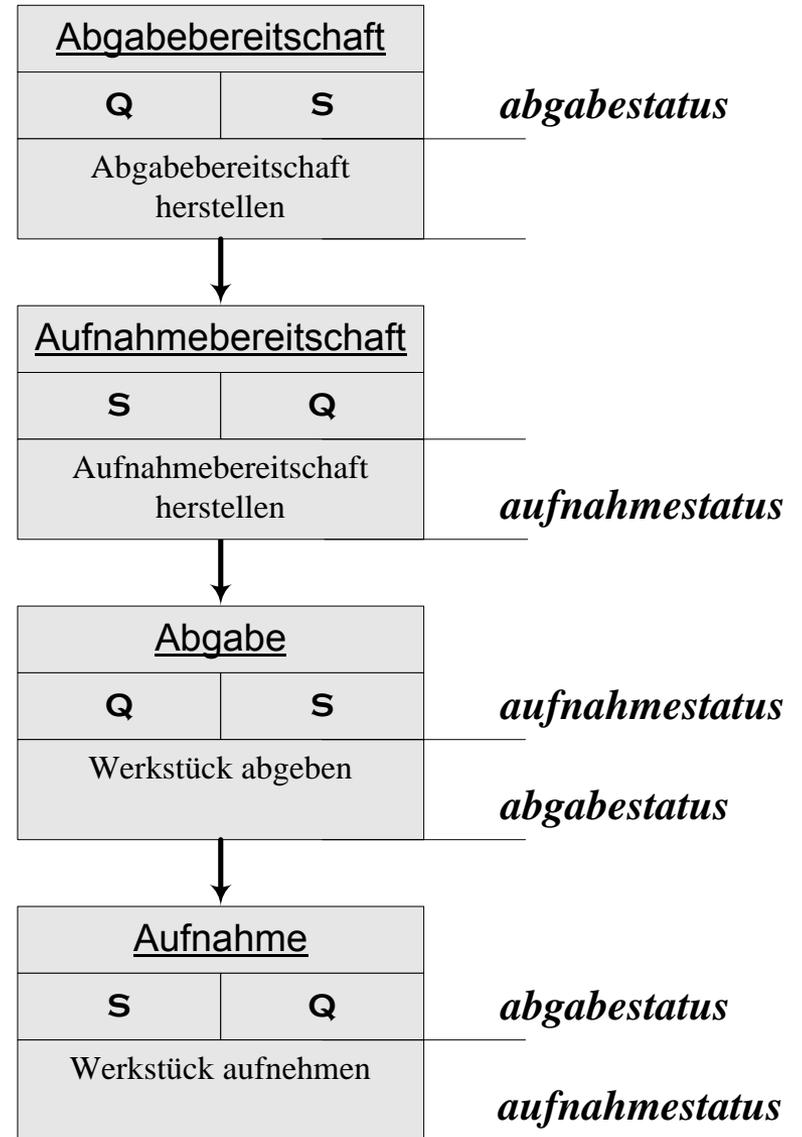
generates $p_el, p_th, p_kin, e_el, e_th, e_kin,$
 $kontotand, opt_param, voranschlag, transportstatus$

Responsibilities:**Liveness:****BANDSTEUERUNG = (Bdiagnose || [Voranschlag · [Transport]])^ω****Safety:**

- $p_el \neq 0 \rightarrow$ Buchen
- $p_el = 0 \rightarrow$ Stornieren
- $(auftrag.dauer > 0) \rightarrow$ Bsetoptimum(auftrag.dauer)
- $(auftrag.kosten > 0) \ \&\& \ (auftrag.dauer \leq 0) \rightarrow$ Bsetoptimum(auftrag.kosten)

Definition des Protokolls Übergabe:

- Teilnehmende Rollen
 - Quelle und Senke
- Initiatorrolle
 - Quelle
- Untergliedert durch einzelne Schritte
- Ein- und Ausgangsvariablen der einzelnen Schritte (des Protokolls)



- [BMG08] Badr, I.; Mubarak, H.; Göhner, P.: Extending the MaSE Methodology for the Development of Embedded Real-Time Systems. Lecture Notes In Artificial Intelligence, Springer, Berlin, 2008
- [BNR06] Bendel, C.; Nestle, D.; Ringelsstein, J.: Bidirectional Energy Management Interface (BEMI) - Integration of the low voltage level into grid communication and control, 19th International conference on Electricity Distribution, Wien, 2007
- [HWH04] Heintel, B.; Wagner, T.; Hohn, G.: Agentenorientiertes dezentrales Energie- und Funktionsmanagement für Kfz. Entwicklerforum Kfz-Elektronik, Ludwigsburg/Stuttgart, Mai 2004
- [Kat09] Katzke, U.: Spezifikation und Anwendung einer Modellierungssprache für die Automatisierungstechnik auf Basis der Unified Modelling Language (UML). Kassel University Press, Kassel, 2009
- [MGW+07] Mubarak, H.; Göhner, P.; Wannagat, A.; Vogel-Heuser, B.: Evaluation of agent oriented methodologies for the development of flexible embedded real-time systems in automation. atp international, Automation Technology in Practice, (5) 1, Oldenbourg Industrieverlag, München, April 2007
- [OJR+05] Oyarzabal, J.; Jimeno, J.; Ruela, J.; Engler, A.; Hardt, C.: Agent based Micro Grid Management System. International Conference on Future Power Systems, Amsterdam, November 2005

- [SBF07] Secchi, C.; Bonfe, M.; Fantuzzi, C.: On the use of UML for modelling mechatronic systems. IEEE Transactions on Automation Science and Engineering, 4 (1), Scottsdale, AZ, USA, 2007, S.105-113
- [Sch09] Schullerer, J.: Effizienz bei Pumpen – Nutzung von Einsparungspotenzialen über die gesamte Lebensdauer. VDI Kongress AUTOMATION 2009, S.55-58, VDI-Verlag, Düsseldorf, Juni 2009
- [ScLo01] Scheutz, M.; Logan, B.: Affective vs. Deliberative Agent Control. Proceedings of the AISB'01 Symposium on Emotion, Cognition and Affective Computing, S. 1–10, UK, York, 2001
- [ScWa09] Schütz, D.; Wannagat, A.: Domänenspezifische Modellierung für automatisierungstechnische Anlagen mit Hilfe der SysML. ATP Automatisierungstechnische Praxis, (51) 3, S. 54-62, Oldenbourg Industrieverlag, München, März 2009
- [Sei09] Seifert, O.: Messung und Überwachung der Energieeffizienz in Hilfskreisläufen zur Kostenreduktion. VDI Kongress AUTOMATION 2009, S.121-124, VDI-Verlag, Düsseldorf, Juni 2009
- [SVB+09] Schütz, D.; Vogel-Heuser, B.; Brabetz, J.; Ayeb, M.: Architekturmodelle zur Bewertung von energetischen Optimierungskriterien – MDA als Bindeglied domänenspezifischer Analyse. VDI Kongress Mechatronik 2009, S. 69 – 76, VDI-Verlag, Düsseldorf, 2009

- [TBV09] Tigges, M.; Bendfeld, J.; Voss, J: Effizientere Nutzung der Offshore-Windenergie durch die Kombination mit Biogaskraftwerken. VDI Kongress AUTOMATION 2009, S.59-64, VDI-Verlag, Düsseldorf, Juni 2009
- [WaVo08a] Wannagat, A.; Vogel-Heuser, B.: Agent oriented software-development for networked embedded systems with real time and dependability requirements the domain of automation. Proceedings of the 17th IFAC World Congress, S. 4144 – 4149, IFAC World Congress, 2008
- [WaVo08b] Wannagat, A.; Vogel-Heuser, B.: Increasing Flexibility and availability of Manufacturing Systems – Dynamic Reconfiguration of Automation Software at Runtime on Sensor Faults. 9th IFAC Workshop on Intelligent Manufacturing Systems, Vortrag Session 5.2, Polen, Stettin, 2008
- [WoJe95] Wooldridge, J.; Jennings, N.: Intelligent agents: Theory and practice. The Knowledge Engineering Review, 10 (2), S. 115 – 152, Cambridge University Press, Cambridge, 1995