

Vergleich des M-Q-M mit Petri-Netzen

Wissenschaftlich orientierte Gegenüberstellung mit konkretem Zuordnungsmodell

Arbeitsnotiz auf Basis der bereitgestellten M-Q-M-Unterlagen und der Petri-Netz-Literatur

7. Mai 2026

Zusammenfassung

Dieses Dokument vergleicht die zuvor entwickelte M-Q-M-Architektur systematisch mit dem Ansatz der Petri-Netze. Der Vergleich erfolgt entlang des Modellkerns, der Zeit- und Hierarchiesemantik, der Simulations- und Verifikationsfähigkeit, der Entscheidungsunterstützung sowie der IT-Architektur. Zugleich wird ein konkretes Zuordnungsmodell $M-Q-M \leftrightarrow \text{Petri-Netz}$ abgeleitet, insbesondere für Places, Transitions, Tokens, Guards und Hierarchie-Module. Das Ergebnis ist asymmetrisch: Petri-Netze sind der reifere Formalismus für Kontrollfluss, Nebenläufigkeit, Aktivierbarkeit und Verifikation; das M-Q-M ist der reichere semantische Rahmen für Zielbild, Regelung, Vollzug, Wirkung und Hypersystem-Kontext. Daraus folgt als präferierte IT-Lösung eine Hybridarchitektur, in der das M-Q-M die Management- und Entscheidungssemantik liefert, während zeitbehaftete hierarchische Coloured Petri Nets den operativen Prozesskern bilden.

Schlagwörter: M-Q-M, Petri-Netze, Coloured Petri Nets, Workflow Nets, Prozessmodellierung, Simulation, Entscheidungsunterstützung, digitale Zwillinge

1 Einleitung

Die Gegenüberstellung vergleicht nicht zwei gleich etablierte Formalismen. Petri-Netze sind ein seit Jahrzehnten breit untersuchter mathematischer Formalismus für diskrete Ereignissysteme, parallele Abläufe und verifizierbare Prozessstrukturen [1, 2, 3, 5]. Das M-Q-M ist demgegenüber in den vorliegenden Unterlagen als projekt- und forschungsnahes Metamodell formuliert, das Aktivitäten statt Objekte in den Mittelpunkt stellt und Prozesse mit Zielbild, Regelwerk, Vollzug, Wirkung, Historie und Kontext koppelt [6, 7, 8, 9].

Die interne Logik des M-Q-M wird in den bereitgestellten Unterlagen über vier Quadranten beschrieben: Q_1 für Transformation, Regelwerk und Parameter, Q_2 für Fluss, Input, Verbrauch und Prozess, Q_3 für Idee, Plan, Budget, Leitbild und Gesamtsystem sowie Q_4 für Outcome, Bestand, Portfolio und Output [6, 7, 8]. Hinzu kommen die Achsen x (Zeit), y (Raum/Menge/Ausdehnung) und z (Modus, Sequenz, Historie) sowie die Hierarchie von Hypersystem, System und Subsystem [6, 7, 8]. In der hier verwendeten mathematischen Rekonstruktion wird dieser Rahmen als stratifizierte Prozesskategorie mit dem Zyklus

$$Q_3 \rightarrow Q_1 \rightarrow Q_2 \rightarrow Q_4 \rightarrow Q_3$$

verstanden [9].

Ziel dieses Papiers ist daher kein Gleichheitsnachweis, sondern eine wissenschaftlich nüchterne Klärung von drei Fragen: Erstens, worin unterscheiden sich beide Ansätze im Modellkern? Zweitens, wo ist Petri-Netz-Theorie klar überlegen und wo ist das M-Q-M semantisch überlegen? Drittens, wie lässt sich das M-Q-M konkret auf Petri-Netz-Konstrukte abbilden, ohne seinen inhaltlichen Mehrwert zu verlieren?

2 Methodische Grundlage des Vergleichs

Für die Petri-Netz-Seite wird auf etablierte Primärliteratur zurückgegriffen. Murata definiert das klassische Petri-Netz als 5-Tupel

$$PN = (P, T, F, W, M_0),$$

mit Places P , Transitions T , Flussrelation F , Kantengewichten W und Anfangsmarkierung M_0 [1]. Dieselbe Quelle beschreibt die bekannte Schaltregel: Eine Transition ist aktiviert, wenn jede Eingangs-Stelle die erforderliche Token-Zahl enthält; beim Feuern werden Tokens aus den Eingangs-Stellen entfernt und in die Ausgangs-Stellen eingefügt [1]. Für datenreiche und hierarchische Modelle werden Coloured Petri Nets herangezogen. Jensen zeigt, dass Tokens dort nicht bloß ununterscheidbare Marken sind, sondern typisierte Werte tragen können; außerdem sind hierarchische Beschreibungen mit wohldefinierten Schnittstellen möglich [2]. Jensen, Kristensen und Wells ergänzen dies um den diskreten Zeitbegriff, modulare Modellstruktur, Simulation, Zustandsraumanalyse, Model Checking und performanzbezogene Auswertung [3]. Für Workflow-Eigenschaften und Korrektheitsanalyse wird auf die Arbeiten von van der Aalst zurückgegriffen, insbesondere auf Workflow Nets und Soundness [4, 5].

Für die M-Q-M-Seite wird ausschließlich auf die vom Nutzer bereitgestellten Projektunterlagen und die darauf aufbauende mathematische Rekonstruktion zurückgegriffen. Diese Unterlagen charakterisieren das Modell ausdrücklich als aktivitätsbasiert statt objektbasiert, führen die vier Quadranten und ihre Achsenbedeutungen aus, beschreiben die z-Achse als historische Zeit bzw. Phasenlogik, definieren geschlossene und offene Systeme sowie die Beziehungen $Q_3 = Q_2 + Q_4$ und $Q_1 + Q_3 = \text{const}$, und verorten den Ansatz in Richtung Analyse, Steuerung und Simulation digitaler Zwillinge [6, 7, 8, 9].

3 Vergleich des Modellkerns

3.1 Was ein Petri-Netz formalisiert

Petri-Netze sind stark, wenn ein System als Menge diskreter Zustände und diskreter Zustandsänderungen verstanden werden kann. Places repräsentieren Bedingungen, Bestände oder Ressourcenlagen; Transitions repräsentieren Ereignisse oder Aufgaben; Tokens kodieren die aktuelle Markierung bzw. Fallzahl; und die Feuervorschrift bestimmt, welche Ereignisse aus einem gegebenen Zustand zulässig sind [1]. Daraus ergeben sich klassische Analysefragen wie Erreichbarkeit, Beschränktheit, Lebendigkeit, Konflikt, Deadlock, Persistenz oder Soundness [1, 5]. In Coloured Petri Nets kommen Datentypen, Arc-Expressions, Guards, Zeitstempel und hierarchische Module hinzu, wodurch sich auch komplexere, datenabhängige und zeitbehaftete diskrete Ereignissysteme präzise modellieren lassen [2, 3].

3.2 Was das M-Q-M formalisiert

Das M-Q-M strukturiert ein System nicht primär über Stellen und Ereignisse, sondern über vier semantisch verschiedene Strata. Q_3 trägt Zielbild, Plan und Identität; Q_1 enthält Regeln, Parameter und Transformationslogik; Q_2 trägt Input, Ressourcen, Rollen, Fluss und Vollzug; Q_4 enthält Ergebnis, Bestand, Portfolio und Wirkung [6, 7, 8]. Diese Strata werden über die Achsen x , y und z beschrieben und in geschlossenen Systemen über die Kopplungen

$$y_1 = y_2, \quad x_2 = x_3, \quad y_3 = y_4, \quad x_4 = x_1$$

verbunden [7]. Zusätzlich treten Bilanz- bzw. Identitätsrelationen auf,

$$Q_3 = Q_2 + Q_4, \quad Q_1 + Q_3 = \text{const},$$

welche im M-Q-M nicht bloß Kontrollfluss, sondern Systemkohärenz ausdrücken [7, 8].

Der zentrale Unterschied liegt damit auf der semantischen Ebene: Ein Petri-Netz beantwortet in erster Linie die Frage, *welche diskreten Schritte aus einem gegebenen Markierungszustand zulässig sind*. Das M-Q-M beantwortet die umfassendere Frage, *welches Ziel verfolgt das System, unter welchen Regeln darf es handeln, wie vollzieht es sich und welche Wirkung entsteht daraus*. Der Preis dieser semantischen Breite ist, dass das M-Q-M bislang nicht denselben standardisierten Analysekanon besitzt wie Petri-Netze.

4 Abstraktionsebene, Zeit und Kontext

4.1 Abstraktion und Semantik

Petri-Netze sind minimalistisch und gerade deshalb analytisch stark. Viele Sachverhalte werden bewusst auf Kontrollfluss, Synchronisation, Konflikt und Ressourcenbelegung reduziert [1, 4]. Das M-Q-M ist semantisch dichter: Es erfasst Zielsystem, Governance, Vollzug und Ergebnis in einem gemeinsamen Rahmen und versteht Aktivitäten ausdrücklich als Entscheidung und Handlung [6, 8]. Dadurch eignet es sich stärker als Management- und Architektursprache, weniger aber als unmittelbarer Ersatz für eine wohletablierte Kontrollfluss-Logik.

4.2 Zeitmodell

Petri-Netze besitzen in ihrer Grundform keine reichhaltige Zeitsemanik; diese wird in zeitbehafteten oder stochastischen Varianten ergänzt. Die Literatur zu Coloured Petri Nets zeigt, dass sich damit Zeitstempel, Verzögerungen, Durchsatz, Warteschlangen, Deadlines und performanzbezogene Fragestellungen modellieren lassen [1, 3]. Das M-Q-M trägt Zeit bereits im Kernmodell: x repräsentiert absolute bzw. relative Zeit, während z als diskrete Achse Phasen, Sequenzen, Entwicklungsstufen und historische Wendepunkte beschreibt [8, 7]. In der Sprache des M-Q-M ist Zeit daher nicht nur ein Attribut von Ereignissen, sondern ein konstitutiver Teil des Systemraums.

4.3 Hierarchie und Kontext

Hierarchische Coloured Petri Nets können große Modelle in Module mit wohldefinierten Schnittstellen zerlegen; Port-/Socket-Places und verwandte Mechanismen koppeln diese Module formal sauber [2, 3]. Das M-Q-M kennt Hierarchie in einem anderen Sinn: nicht nur als technische Zerlegung, sondern als Einbettung in Hypersystem, System und Subsystem. Das Hypersystem liefert Kontext, Logistik, Kommunikationsmittel oder externe Treiber; Subsysteme bilden die Bestandteile des Systems; und in Wertschöpfungsketten kann das Ergebnis Q_4 eines Systems zum Input Q_2 des nächsten werden [6, 7, 8]. Diese Kontextsättigung ist für Management- und Digital-Twin-Szenarien besonders wertvoll, analytisch aber weniger standardisiert als die modulare Komposition in Coloured Petri Nets.

5 Simulation, Entscheidung und Verifikation

5.1 Simulation

Beide Ansätze eignen sich zur Simulation, aber mit anderem Schwerpunkt. Coloured Petri Nets sind eine diskrete Ereignissprache; sie erlauben ausführbare Modelle, Zustandsraumanalyse, Modellprüfung und performanzbezogene Experimente [3]. Das M-Q-M ist hingegen von Hause aus auf die Kombination von Beschreibung, Simulation und Entscheidungsunterstützung ausgerichtet, insbesondere im Kontext digitaler Zwillinge und grüner virtueller Fabriken [6, 7]. Für operative Prozesssimulation ist die Petri-Netz-Seite methodisch reifer; für szenariobasierte Führungssimulation mit Zielbild, Regelwerk und Wirkung ist das M-Q-M semantisch reicher.

5.2 Entscheidungsunterstützung

Hier zeigt sich der größte strukturelle Unterschied. Die Workflow-Literatur betont, dass Petri-Netz-basierte Verifikation oft vom Datenraum abstrahiert und viele reale Entscheidungen als Routing-Wahl oder als von Menschen bzw. komplexen Anwendungen gesetzte Kontrollinformation behandelt [4, 5]. Das ist kein Mangel, sondern Teil der methodischen Fokussierung auf Kontrollfluss. Das M-Q-M dagegen enthält eine explizite Entscheidungsschicht: Q_3 formuliert Ziel und Orientierung, Q_1 prüft Regeln und Freigaben, Q_2 repräsentiert die operative Machbarkeit und Q_4 die Wirkung. Dadurch eignet sich das M-Q-M besonders als Entscheidungsrahmen für Simulation und Management.

5.3 Verifikation

Wenn die Leitfrage lautet, ob ein Ablauf korrekt, deadlock-frei, beschränkt, sound oder frei von problematischen Kontrollflussmustern ist, sind Petri-Netze klar überlegen. Für klassische Netze und ihre Unterklassen existiert ein ausgereifter Katalog an Eigenschaften und Analyseverfahren [1]. Für Workflow Nets zeigt van der Aalst unter anderem, dass Soundness bei freien Wahlstrukturen effizient entscheidbar ist und dass freie Wahlstrukturen ein guter Kompromiss zwischen Ausdruckskraft und Analysierbarkeit sind [5]. Das M-Q-M besitzt in der vorliegenden Fassung zwar mathematische Invarianten und Kopplungsbedingungen, aber noch keinen vergleichbar etablierten Kanon allgemeiner Verifikationsalgorithmen. Seine Stärke liegt deshalb eher

in semantischer Vollständigkeit als in klassischer formaler Prüfbarkeit.

6 Verdichtete Gegenüberstellung

Vergleichsdimension	Petri-Netze	M-Q-M
Modellkern	Stellen, Transitionen, Marken und Feuervorschriften; stark für diskrete Zustandswechsel, Synchronisation und Nebenläufigkeit [1].	Vier semantische Strata Q_3, Q_1, Q_2, Q_4 mit Zielbild, Regelung, Vollzug und Wirkung sowie Achsen x, y, z [6, 7, 8].
Semantischer Schwerpunkt Zeitmodell	Kontrollfluss, Aktivierbarkeit, Konflikt, Parallelität, Ressourcennutzung. In der Grundform unzeitlich; Zeit und Stochastik in Erweiterungen.	Zielkohärenz, Governance, Prozess, Ergebnis, Historie und Kontext. Zeit im Kern über x und diskrete Entwicklungs-/Phasenlogik über z [8, 7].
Datenmodell	In PT-Netzen minimal; in Coloured Petri Nets reichhaltig über Color Sets, Guard-Ausdrücke und Arc-Expressions [2, 3].	Von Anfang an semantisch datenhaltig, da jeder Quadrant eigene Zustandsattribute trägt [6, 7].
Hierarchie	Formal modulare Komposition hierarchischer Netze mit wohldefinierten Interfaces [2, 3].	Kontextuelle Einbettung in Hypersystem, System und Subsystem [6, 7, 8].
Entscheidungen	Typischerweise indirekt als Routing, Guard oder nichtdeterministische Wahl.	Explizit im Modellkern: Ziel, Regel, Vollzug und Wirkung bilden gemeinsam den Entscheidungsraum.
Verifikation	Stark; Reachability, Boundedness, Liveness, Soundness, Model Checking [1, 5, 3].	Derzeit stärker über Invarianten und Plausibilitätsbeziehungen als über standardisierte globale Verifikation.
Erklärbarkeit für Management	Eher mittel; sehr präzise, aber semantisch sparsam.	Hoch; besonders stark für Strategie, Governance und Wirkungsbegründung.
Stärke in IT-Architekturen	Prozesskern, Ausführbarkeitsraum, formale Prüfbarkeit.	Führungs- und Entscheidungsrahmen, Semantiksicht, Digital-Twin- und Management-Integration.

7 Konkretes Zuordnungsmodell M-Q-M \leftrightarrow Petri-Netz

7.1 Grundsatz der Abbildung

Eine direkte 1:1-Abbildung *Quadrant = einzelne Transition* wäre zu grob und würde die Semantik des M-Q-M zerstören. Die tragfähigere Abbildung ist zweistufig:

- 1.) Ein M-Q-M-System wird als *hierarchisches, zeitbehaftetes Coloured Petri Nets* modelliert.
- 2.) Die vier Quadranten werden nicht auf vier isolierte Elemente reduziert, sondern auf *Stellenfamilien plus Kopplungsübergänge* abgebildet.

Formal lautet das empfohlene Top-Level-Skelett:

$$P = P_3 \cup P_1 \cup P_2 \cup P_4, \quad T = \{\tau, \pi, \varepsilon, \rho\} \cup T_{\text{lokal}}.$$

Dabei bildet

$$Q_3 \xrightarrow{\tau} Q_1 \xrightarrow{\pi} Q_2 \xrightarrow{\varepsilon} Q_4 \xrightarrow{\rho} Q_3$$

den strategisch-regulatorisch-operativen Rückkopplungszyklus ab.

7.2 Places

Places tragen im Petri-Netz den persistierenden oder beobachtbaren Zustand. Im M-Q-M sind das vor allem die semantischen Bestandsräume der vier Quadranten. Deshalb werden Quadranten als *Stellenfamilien* modelliert:

$$P_3 = \{Goal, Scenario, Budget, Identity\},$$

$$P_1 = \{PolicyActive, ConstraintSet, Approval, Thresholds\},$$

$$P_2 = \{Ready, Running, Waiting, ResourceFree, ResourceBusy\},$$

$$P_4 = \{Output, Stock, KPI, Evidence\}.$$

Die Bedeutung ist klar: Q_3 wird zu Plan- und Identitäts-Stellen, Q_1 zu Policy- und Regel-Stellen, Q_2 zu Prozess- und Ressourcen-Stellen, Q_4 zu Ergebnis- und Evidenz-Stellen. Das passt sowohl zur M-Q-M-Semantik als auch zur Interpretation von Places als Bedingungen, Bestände oder Ressourcenlagen [1, 6, 7, 8].

7.3 Transitions

Transitions repräsentieren Zustandsänderungen oder Ereignisse. Auf der Top-Level-Ebene ergeben sich direkt aus der M-Q-M-Architektur vier Grundtransitionen:

$$\tau = \text{CompilePlanToPolicy}, \quad \pi = \text{ReleaseToExecution}, \quad \varepsilon = \text{CompleteExecution}, \quad \rho = \text{FeedbackAndReplan}.$$

Hinzu treten lokale Prozess-Transitionen im Bereich Q_2 , etwa *StartTask*, *AllocateResource*, *Synchronize*, *FinishTask*. Petri-netz-theoretisch bleiben dies ganz normale Transitionen; M-Q-M-semantisch zerfallen sie aber in einen geregelten Übergang von Zielbild über Governance zum Vollzug und zur Wirkung.

7.4 Tokens

Tokens repräsentieren im M-Q-M nicht die Quadranten selbst, sondern konkrete Aktivitäts- bzw. Fallinstanzen. Für gewöhnliche PT-Netze wären dies ununterscheidbare schwarze Marken. Für eine inhaltlich angemessene Abbildung sind jedoch farbige, typisierte Tokens erforderlich. Ein minimales Token-Schema lautet:

$$\text{CaseToken} = (id, sys, x, y, z, planRef, policyRef, res, outcome, prov).$$

Zusätzliche spezialisierte Token-Typen können *PlanToken*, *PolicyToken*, *ResourceToken*, *OutcomeToken* oder *DriverToken* sein. Die Felder x , y und z tragen dabei die M-Q-M-Achsensemantik; *planRef* und *policyRef* koppeln eine Aktivitätsinstanz an Q_3 und Q_1 ; *outcome* und *prov* (Provenienz) ermöglichen Wirkungsauswertung und Audit. Erst in dieser Form wird die Petri-Netz-Seite wirklich M-Q-M-fähig.

7.5 Guards

Guards sind der natürliche Ort für die operative Ausprägung von Q_1 . Das statische Regelregime lebt in Q_1 -Places; die Frage, ob ein konkreter Übergang in einer konkreten Situation zulässig ist, lebt als Boolescher Ausdruck auf der Transition. Allgemein:

$$G(t) = \text{BoolExpr}(\textit{Plan}, \textit{Policy}, \textit{Phase}, \textit{Resource}, \textit{KPI}).$$

Ein typischer Freigabe-Guard für den Übergang von Q_1 nach Q_2 wäre beispielsweise:

$$G(\pi) = \textit{policy.active} \wedge \textit{budget} \geq \textit{cost} \wedge z \in \textit{allowedPhases} \wedge \textit{resource.free} \geq \textit{demand}.$$

Auch M-Q-M-Kopplungen lassen sich so operationalisieren, etwa

$$(y_1 = y_2) \wedge (x_2 = x_3) \wedge (y_3 = y_4) \wedge (x_4 = x_1),$$

oder als Bindungsgleichheit zwischen Variablen mehrerer Tokens. Die globalen Beziehungen $Q_3 = Q_2 + Q_4$ und $Q_1 + Q_3 = \text{const}$ sollten nicht als rohe Netzstruktur, sondern als Monitor-Regeln oder Beobachter-Transitionen modelliert werden.

7.6 Hierarchie-Module

Für die Hierarchie ist die Abbildung besonders klar. Ein Hypersystem wird als Kontext- bzw. Umweltmodul modelliert, ein System als Top-Level-Modul und Subsysteme als verschachtelte Teilmodule. In Coloured Petri Nets geschieht dies über hierarchische Module mit wohldefinierten Interfaces [2, 3]. Die empfohlene Struktur lautet:

$$\textit{HS_Context} \supset \textit{System_S} \supset \{ \textit{Q3_Planning}, \textit{Q1_Policy}, \textit{Q2_Process}, \textit{Q4_Outcome} \} \supset \textit{Subsystem}_{1..n}.$$

Für verknüpfte Systeme bzw. Wertschöpfungsketten gilt dann das Interface-Schema

$$P_{4,\text{out}}^{(n)} \iff P_{2,\text{in}}^{(n+1)},$$

also: Das Ergebnis des einen Systems wird zum Prozess-Input des nächsten. Genau das entspricht der M-Q-M-Idee, dass Q_4 eines Systems in Q_2 des Folgesystems übergeht [7, 8, 6].

8 Interpretation des Hybridmodells

Die Abbildung zeigt, dass Petri-Netze das M-Q-M nicht sinnvoll ersetzen, wohl aber dessen operative Schicht präzisieren können. Daraus ergibt sich ein dreistufiges Architekturprinzip:

- 1.) **Q₃-Ebene:** Zielbild, Szenario, Plan, Budget und Prioritäten.
- 2.) **Q₁-Ebene:** Policies, Regeln, Schwellenwerte, Freigaben, Compliance.
- 3.) **Q₂-Ebene:** eigentlicher Ablaufkern als hierarchisches, zeitbehaftetes Coloured Petri Nets.
- 4.) **Q₄-Ebene:** Ergebnis, Evidenz, KPI, Wirkung, Re-Planungsimpulse.

Damit ergibt sich eine klare Arbeitsteilung: Das Petri-Netz definiert den *Ausführbarkeitsraum* der zulässigen diskreten Schritte. Das M-Q-M definiert den *Bedeutungs- und Entscheidungsraum* darüber. Anders formuliert: Petri-Netze beantworten die Frage, *was aus Sicht von Kontrollfluss und Ressourcen jetzt feuern darf*; das M-Q-M beantwortet zusätzlich, *welcher dieser zulässigen Schritte im Sinne von Zielbild, Regeln, operativer Machbarkeit und erwarteter Wirkung vorzuziehen ist*.

Gerade für eine IT-Plattform zur Beschreibung, Simulation und Entscheidungsunterstützung ist diese Trennung fruchtbar. Die Petri-Netz-Schicht liefert formale Strenge und Verifikation; die M-Q-M-Schicht liefert Management-Semantik, Erklärbarkeit und Anschluss an digitale Zwillinge.

9 Schlussfolgerung

Der Vergleich führt zu einem klaren, aber nicht polarisierenden Ergebnis. Petri-Netze sind dem M-Q-M dort überlegen, wo Prozesse als diskrete Ereignissysteme formal prüfbar, ausführbar und auf Eigenschaften wie Erreichbarkeit, Beschränktheit, Deadlock-Freiheit oder Soundness untersucht werden sollen. Das M-Q-M ist Petri-Netzen dort überlegen, wo Prozesse nicht nur als Kontrollfluss, sondern als zielorientierte, regelgebundene, kontextabhängige und wirkungsbezogene Systeme verstanden werden sollen.

Die praktisch stärkste Synthese ist daher keine Entweder-oder-Entscheidung, sondern eine Schichtung: M-Q-M als übergeordnete Management-, Semantik- und Entscheidungsschicht; hierarchische zeitbehaftete Coloured Petri Nets als formaler Prozess- und Simulationskern. In dieser Form ergänzen sich beide Ansätze nicht nur, sondern kompensieren wechselseitig ihre Grenzen.

Literatur

- [1] T. Murata. Petri Nets: Properties, Analysis and Applications. *Proceedings of the IEEE*, 77(4):541–580, 1989. DOI: 10.1109/5.24143.
- [2] K. Jensen. An Introduction to the Theoretical Aspects of Coloured Petri Nets. In J. W. de Bakker, W.-P. de Roever und G. Rozenberg (Hrsg.), *A Decade of Concurrency*, Lecture Notes in Computer Science 803, S. 230–272. Springer, 1994.
- [3] K. Jensen, L. M. Kristensen und L. Wells. Coloured Petri Nets and CPN Tools for Modelling and Validation of Concurrent Systems. *International Journal on Software Tools for Technology Transfer*, 9(3/4):213–254, 2007. DOI: 10.1007/s10009-007-0038-x.
- [4] W. M. P. van der Aalst. The Application of Petri-Nets to Workflow Management. *Journal of Circuits, Systems and Computers*, 8(1):21–66, 1998. DOI: 10.1142/S0218126698000043.

- [5] W. M. P. van der Aalst. Workflow Verification: Finding Control-Flow Errors Using Petri-Net-Based Techniques. In W. M. P. van der Aalst, J. Desel und A. Oberweis (Hrsg.), *Business Process Management: Models, Techniques, and Empirical Studies*, Lecture Notes in Computer Science 1806, S. 161–183. Springer, 2000. DOI: 10.1007/3-540-45594-9_11.
- [6] *M-Q-M_Projektstatus_260212.pdf*. Unveröffentlichte Projektunterlage, vom Nutzer bereitgestellt.
- [7] *Das Multi-Quadranten-Modell.docx*. Unveröffentlichte Projektunterlage, vom Nutzer bereitgestellt.
- [8] *Glossar des M-Q-M.docx*. Unveröffentlichte Projektunterlage, vom Nutzer bereitgestellt.
- [9] *Mathematische Vorlage Agentisches System.docx*. Unveröffentlichte Projektunterlage, vom Nutzer bereitgestellt.