

## **BERECHNUNG VON BAUABLÄUFEN MIT VERSCHIEDENEN AUSFÜHRUNGSVARIANTEN**

**M. König\* and E. Tauscher**

*\*Bauhaus-University Weimar  
Marienstraße 7, D-99423 Weimar  
E-mail: markus.koenig@uni-weimar.de*

**Keywords:** Ausführungsvarianten, Generierung von Bauabläufen, Workflow-Netze

**Abstract.** *Prozesse im Bauingenieurwesen sind komplex und beinhalten eine große Anzahl verschiedener Aufgaben mit vielen logischen Abhängigkeiten. Basierend auf diesen projektspezifischen Abhängigkeiten wird gewöhnlich ein Bauablaufplan manuell erstellt. In der Regel existieren mehrere Ausführungsvarianten und somit alternative Bauabläufe um ein Projekt zu realisieren. Welche dieser Ausführungsvarianten zur praktischen Anwendung kommt, wird durch den jeweiligen Projektmanager bestimmt. Falls Änderungen oder Störungen während des Bauablaufs auftreten, müssen die davon betroffenen Aufgaben und Abläufe per Hand modifiziert und alternative Aufgaben sowie Abläufe stattdessen ausgeführt werden. Diese Vorgehensweise ist oft sehr aufwändig und teuer. Aktuelle Forschungsansätze beschäftigen sich mit der automatischen Generierung von Bauabläufen. Grundlage sind dabei Aufgaben mit ihren erforderlichen Voraussetzungen und erzeugten Ergebnissen. Im Rahmen dieses Beitrags wird eine Methodik vorgestellt, um Bauabläufe mit Ausführungsvarianten in Form von Workflow-Netzen zu jeder Zeit berechnen zu können. Die vorgestellte Methode wird anhand eines Beispiels aus dem Straßenbau schematisch dargestellt.*

# 1 EINLEITUNG

Für die Ausführung von Bauprojekten wird eine Vielzahl von Aufgaben definiert. Die Projektaufgaben werden anschließend in eine logische Reihenfolge gebracht. Dadurch entsteht genau ein gültiger Projektablauf für die Ausführung eines Bauvorhabens. Die Festlegung der logischen Reihenfolge der Aufgaben erfolgt in der Regel manuell. Besonders bei großen Bauprojekten sind verschiedene Ausführungsvarianten zur Erreichung einer Teilleistung möglich. Bei der Aufstellung des Ablaufs wird jedoch häufig nur eine Variante berücksichtigt. Welche Varianten für die Ausführung sinnvollerweise zu bearbeiten sind, kann sich im Laufe des Projektfortschrittes auf Grund von neuen Randbedingungen ändern. Ändern sich Randbedingungen, müssen eventuell andere oder ganz neue Ausführungsvarianten zur Erbringung der Teilleistungen geplant werden. Eine manuelle Anpassung des geplanten Projektablaufes ist notwendig geworden. Die Anpassungen von Ablaufplänen sind jedoch nicht trivial und erfordern einen nicht unerheblichen Zeit- und Ressourcenaufwand.

Aktuelle Forschungsansätze beschäftigen sich mit der Berechnung von Bauabläufen [1][2][3]. Grundlage sind dabei genau definierte Aufgaben allen erforderlichen Voraussetzungen (Inputs) und erzeugten Ergebnissen (Outputs). Bei diesen Ansätzen wird vorausgesetzt, dass keine zwei Aufgaben gleiche Ergebnisse erzeugen. Ein wesentlicher Vorteil ist, dass die Aufgaben mit den entsprechenden Bedingungen unabhängig voneinander definiert werden können. Bei der Generierung des Ablaufs kann anschließend auf der Grundlage der Bedingungen geprüft werden, ob alle Aufgaben korrekt bearbeitet werden können.

$$\begin{array}{ll}
 C & \text{Menge der Bedingungen} \\
 T & \text{Menge der Aufgaben} \\
 I \subseteq C \times T & \text{Relation zwischen Bedingungen und Aufgaben (Inputs)} \\
 O \subseteq T \times C & \text{Relation zwischen Aufgaben und Bedingungen (Outputs)} \\
 \text{mit } \bigwedge_{c \in C} (|O \circ c| \leq 1) & \quad \quad \quad (1)
 \end{array}$$

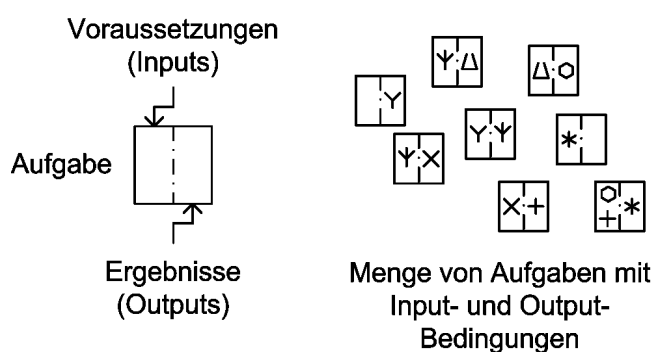


Figure 1: Input- und Output-Bedingungen von Aufgaben

Mit Hilfe der Relationenalgebra kann eine zugehörige Aufgabenabfolge wie folgt berechnet werden (siehe Abbildung 2):

$$S = O \circ I \quad \text{Relation der Aufgabenabfolge} \quad (2)$$

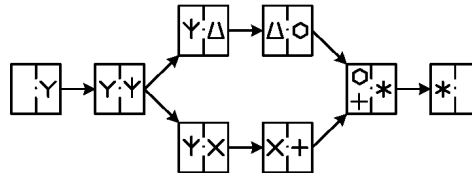


Figure 2: Generierte Aufgabenabfolge

Im Rahmen dieses Beitrages wird der Ansatz verfolgt, verschiedene Ausführungsvarianten bei der Generierung von Bauabläufen zu berücksichtigen. Die frühzeitige Berücksichtigung von Ausführungsvarianten hat den Vorteil, dass auf Änderungen oder Störungen im Projekt sehr schnell und geplant reagiert werden kann. Durch die Festlegung von mehreren Ausführungsvarianten besteht auch die Möglichkeit, dass Ablaufszenarien mit unterschiedlichen Randbedingungen analysiert werden können. Um den Planer bei der Auswahl von Ausführungsvarianten zu unterstützen, soll ein Konzept für die Bewertung von Varianten umgesetzt werden. Die Entscheidung für bestimmte Varianten wird dadurch nachvollziehbar dokumentiert. Dies ist besonders bei nachträglichen Änderungen von besonderer Bedeutung.

## 2 GENERIERUNG VON BAUABLÄUFEN

Abläufe mit mehreren Ausführungsvarianten können nicht mit Hilfe einer Relation  $S$  bzw. eines schlichten Graphen  $G = (T, S)$  beschrieben werden. In der Literatur sind verschiedene Modelle definiert, um Entscheidungssituation und somit Alternativen in einem Ablauf abzubilden. In der Netzplantechnik werden zum Beispiel Netze mit einfachen oder stochastischen Entscheidungsknoten verwendet. Für solche Entscheidungsknoten existieren jedoch keine einheitlichen mathematischen Formulierungen. Für die Beschreibung, Simulation und formale Analyse von beliebigen Abläufen haben sich Petri-Netze als besonders geeignet herausgestellt. Durch die Verwendung von zwei verschiedenen Knotenmengen können parallele sowie alternative Abläufe formal definiert werden. Auf der Basis von Petri-Netzen wurden später so genannte Workflow-Netze entwickelt [4]. Ein Workflow-Netz beschreibt einen Ablauf mit genau einem Startknoten und genau einem Endknoten. Es wird gefordert, dass jeder Knoten von dem Startknoten und der Endknoten von jedem anderen Knoten aus erreichbar sein muss. Parallele Teilabläufe werden durch AND-Splits gestartet und durch AND-Joins wieder zusammengeführt. Alternative Teilabläufe werden durch XOR-Splits gestartet und durch XOR-Joins synchronisiert.

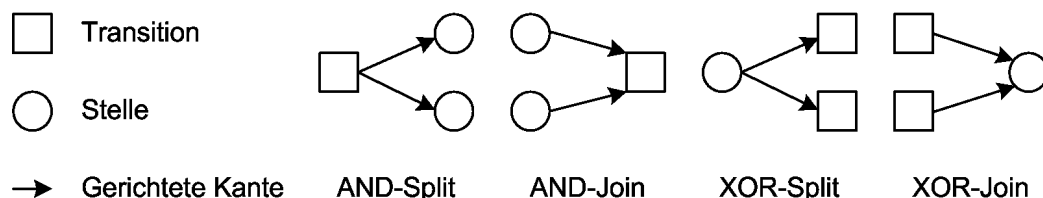


Figure 3: Splits und Joins für Workflow-Netze

Für die Analyse von Abläufen sind mathematische Verfahren auf der Basis von Workflow-

Netzen in der Literatur definiert [4][5]. Die hier vorgestellte automatische Generierung von Bauabläufen mit verschiedenen Ausführungsvarianten liefert als Ergebnis ein Workflow-Netz.

Aufgaben zu verschiedenen Ausführungsvarianten liefern gleiche Ergebnisse. Die Relation  $O$  ist daher keine linkseindeutige  $1 : n$  Relation mehr sondern eine allgemeine  $m : n$  Relation.

$$O \subseteq T \times C \quad \text{Relation zwischen Aufgaben und Bedingungen (Outputs)} \quad (3)$$

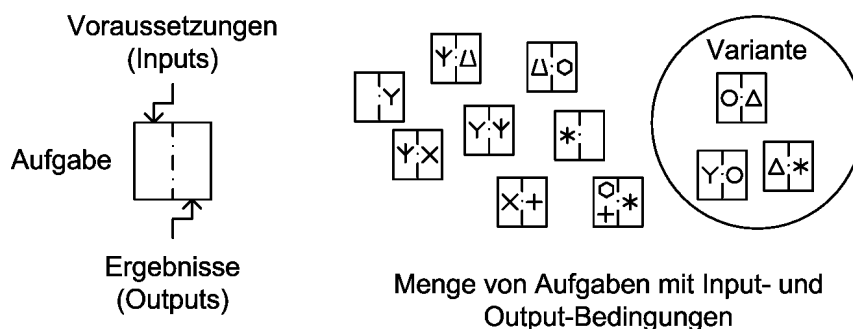


Figure 4: Aufgaben mit Input- und Output-Bedingungen für verschiedene Varianten

Eine sinnvolle Festlegung der Aufgaben und der zugehörigen Bedingungen ist für die Generierung von Workflow-Netzen erforderlich. Zur Bearbeitung einer Aufgabe werden alle Input-Bedingungen benötigt. Während der Bearbeitung werden für jede Aufgabe alle definierten Ergebnisse (Output-Bedingungen) erzeugt. Für die Generierung eines Workflow-Netzes werden als erstes die XOR-Joins, XOR-Splits, AND-Joins und AND-Splits identifiziert. Eine Output-Bedingung, die ein Ergebnis von mehr als einer Aufgabe ist, kennzeichnet einen XOR-Join.

$$xorJoins := \{c \in C \mid |O \circ c| > 1\} \quad (4)$$

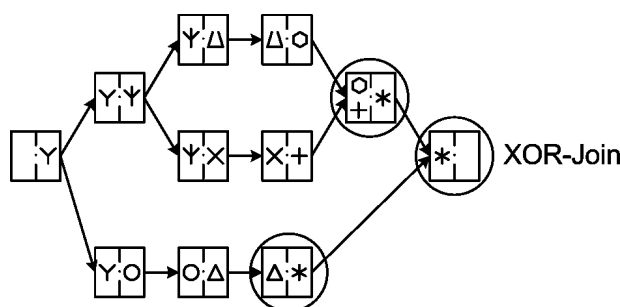


Figure 5: Identifizierung von XOR-Joins

Zu jedem XOR-Join sind im nächsten Schritt die zugehörigen XOR-Splits zu ermitteln. Für die Bestimmung von XOR-Splits gilt, dass jedem XOR-Join mindestens ein XOR-Split zugeordnet ist. Falls die Bedingung nicht erfüllt ist, definieren die vorhandenen Aufgaben und Bedingungen keinen korrekten Bauablauf. Zu einem XOR-Join  $x$  werden die Bedingungen  $c$

als XOR-Splits definiert, für die mindestens zwei disjunkte Kantenzüge von  $c$  nach  $x$  in der Kantenmenge  $I \cup O$  existieren.

$$\begin{aligned} xorSplits(x) &:= \{c \in C \mid |W(c, x)| > 1\} \\ W(c, x) &\quad \text{Menge der disjunkten Kantenzüge von } c \text{ nach } x \end{aligned} \quad (5)$$

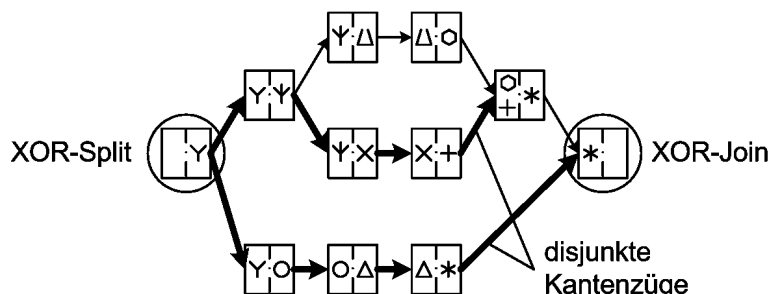


Figure 6: Identifizierung von XOR-Splits

Zur Ermittlung der parallelen Abläufe werden die Aufgaben untersucht. Besitzt eine Aufgabe mehr als eine Voraussetzung, dann wird die entsprechende Aufgabe als AND-Join identifiziert.

$$andJoins := \{t \in T \mid |I^T \circ t| > 1\} \quad (6)$$

Analog zur Ermittlung der XOR-Splits werden zu jedem AND-Join die passenden AND-Splits bestimmt. Für die Bestimmung von AND-Splits gilt, dass jedem AND-Join mindestens ein AND-Split zugeordnet ist. Falls die Bedingung nicht erfüllt ist, definieren die vorhandenen Aufgaben und Bedingungen keinen korrekten Bauablauf. Zu einem AND-Join  $y$  werden die Aufgaben  $t$  als AND-Splits definiert, für die mindestens zwei disjunkte Kantenzüge von  $t$  nach  $y$  in der Kantenmenge  $I \cup O$  existieren.

$$\begin{aligned} andSplits(y) &:= \{t \in T \mid |W(t, y)| > 1\} \\ W(t, y) &\quad \text{Menge der kantendisjunkten Kantenzüge von } t \text{ nach } y \end{aligned} \quad (7)$$

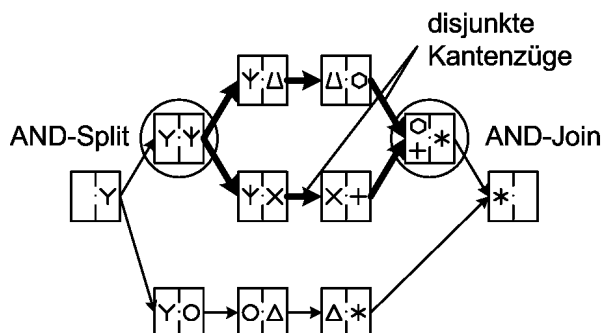


Figure 7: Identifikation von AND-Joins und AND-Splits

Nach der Kennzeichnung aller Splits und Joins kann jetzt auf Grundlage der vorhandenen Aufgaben und Bedingungen ein Workflow-Netz formal erstellt werden. Das Workflow-Netz

$N$  besteht aus einer Menge von Transitionen  $T$ , einer Menge von Stellen  $S$  und einer binären Relation  $F$  über  $S \cup T$ .

$$\begin{aligned}
 N &= (S, T, F) \quad \text{mit} \\
 S \cap T &= \{\}, \\
 S \cup T &\neq \{\} \quad \text{und} \\
 F &\subseteq (S \times T) \cup (T \times S)
 \end{aligned} \tag{8}$$

Jede Aufgabe definiert genau eine Transition im Workflow-Netz. Die Stellen werden auf der Grundlage der vorhandenen Bedingungen erstellt. Der folgende Algorithmus kann für die Generierung des Workflow-Netzes verwendet werden:

1. Die Menge der Aufgaben  $T$  wird als Menge der Transitionen  $T$  definiert.

$$T \quad \text{Menge der Transitionen} \tag{9}$$

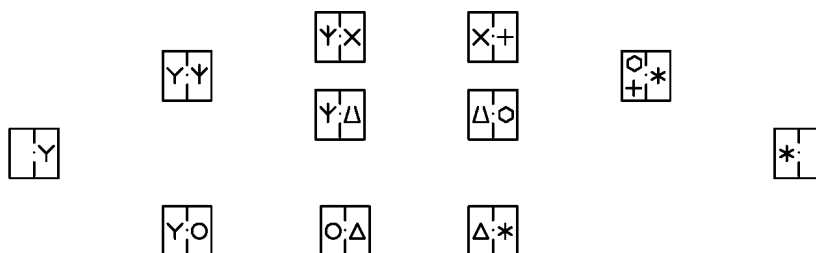


Figure 8: Eingefügte Menge der Transitionen

2. Die XOR-Splits und XOR-Joins werden in die Menge der Stellen  $S$  eingefügt. Es wird jeweils eine Kante zwischen einem XOR-Split/Join  $s$  und einer Aufgabe  $t$  definiert, wenn  $s$  eine Input-Bedingung von  $t$  ist sowie eine Kante zwischen einer Aufgabe  $t$  definiert und einem XOR-Split/Join  $s$ , wenn  $s$  eine Output-Bedingung von  $t$  ist.

$$\begin{aligned}
 S &= S \cup \text{xorSplits} \cup \text{xorJoins} \\
 F &= F \cup \{(s, t) \in I \mid s \in \text{xorSplits} \vee s \in \text{xorJoins}\} \\
 F &= F \cup \{(t, s) \in O \mid s \in \text{xorSplits} \vee s \in \text{xorJoins}\}
 \end{aligned} \tag{10}$$

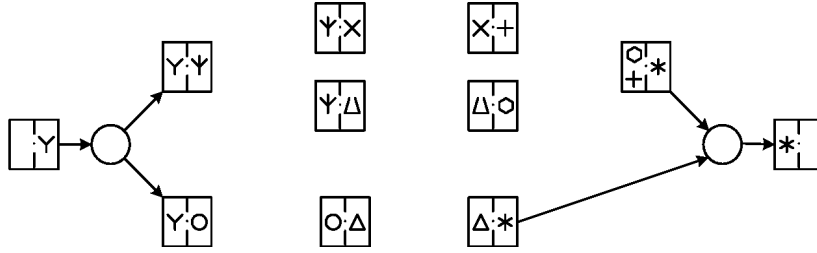


Figure 9: Eingefügte XOR-Splits und XOR-Joins mit zugehörigen Kanten

3. Im nächsten Schritt werden die Stellen und Kanten zwischen parallelen und sequentiellen Aufgaben in das Workflow-Netz eingefügt. Es wird eine neue Stelle  $s_{x,y}$  und zwei Kanten  $(x, s_{x,y})$  und  $(s_{x,y}, y)$  eingefügt, wenn eine Kante  $(x, y)$  in der Kantenmenge  $O_{\setminus xor} \circ I_{\setminus xor}$  existiert. Die Relationen  $O_{\setminus xor}$  und  $I_{\setminus xor}$  sind beschränkte Relationen  $O$  und  $I$  bzgl. aller Bedingungen, die keine XOR-Splits oder XOR-Joins sind.

$$\begin{aligned}
 S &= S \cup \{new\ s_{x,y} \mid (x, y) \in O_{\setminus xor} \circ I_{\setminus xor}\} \\
 F &= F \cup \{new\ (x, s_{x,y}) \mid (x, y) \in O_{\setminus xor} \circ I_{\setminus xor}\} \\
 &\quad \cup \{new\ (s_{x,y}, y) \mid (x, y) \in O_{\setminus xor} \circ I_{\setminus xor}\}
 \end{aligned}$$

$$\begin{aligned}
 \text{mit } O_{\setminus xor} &= O \mid (C \setminus (xorSplits \cup xorJoins)) \quad \text{und} \\
 I_{\setminus xor} &= I \mid (C \setminus (xorSplits \cup xorJoins))
 \end{aligned} \tag{11}$$

4. Im letzten Schritt wird eine Startstelle  $s_s$  sowie eine Endstelle  $s_e$  mit den entsprechenden Kanten in das Workflow-Netz eingefügt.

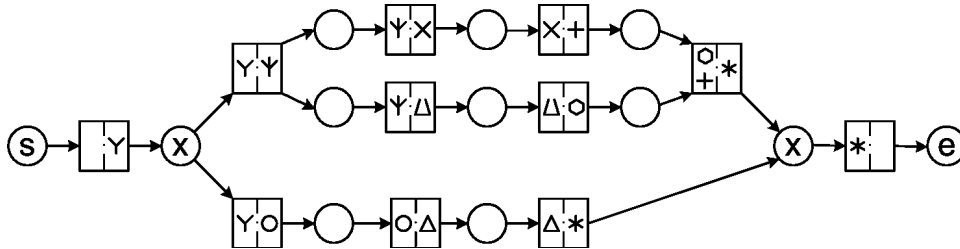


Figure 10: Generiertes Workflow-Netz

### 3 ANWENDUNGSBEISPIEL

Die automatische Generierung von Workflow-Netz auf der Grundlage von Aufgaben und Bedingungen mit mehreren Ausführungsmöglichkeiten wird anhand eines Beispiels aus dem Straßenbau exemplarisch dargestellt [6].

Es werden drei verschiedene Varianten für die Erstellung einer Asphaltdeckschicht modelliert. Im Einzelnen handelt es sich hierbei um Gussasphalt, Splittmastixasphalt und Offener Asphalt. Alle drei Varianten unterscheiden sich hinsichtlich ihrer Eigenschaften, ihrer Herstellung und ihrem Einsatz. Gussasphalt ist überall einsetzbar. Typische Einsatzgebiete sind Straßen jeden Typs. Beim Einsatz auf Brücken besitzt Gussasphalt den Vorteil, dass die beim

Einbau notwendige Verdichtung ohne Vibration möglich ist. Offenporiger Asphalt zeichnet sich durch guten Wasserabfluss infolge der vorhandenen Poren aus und mindert somit die Gefahr des Aquaplaning. Dafür muss jedoch die Schicht unterhalb der Deckschicht abgedichtet und eine Entwässerung neben der Fahrbahn sichergestellt werden. Ein weiterer Effekt ist eine Minderung der Fahrgeräusche ab einer Geschwindigkeit von ca. 60 km/h. Für den Einbau von offenporigem Asphalt werden jedoch spezielle Fertiger benötigt. Splittmastix Asphalt wird heutzutage für fast alle Schwerlastverkehrsstraßen verwendet. Er enthält Zusatzstoffe (z.B. Zellulose), die dem Splitt und Bitumen zugesetzt werden. Die Rissbildung in der Deckschicht wird dadurch vermindert.

In der Abbildung 11 sind die Aufgaben zur Erstellung der Asphalttrageschicht für Schnellverkehrsstraßen aufgeführt. Die Voraussetzungen und Ergebnisse sowie die Zuordnung zu Aufgaben sind in den Abbildungen 12 und 13 dargestellt. Auf der Grundlage dieser Aufgaben und Bedingungen werden die Joins und Splits identifiziert. In diesem Beispiel sind die Bedingungen C03 und C04 die XOR-Splits, die Bedingungen C11 und C16 die XOR-Joins, die Aufgaben A20, A21 und A22 die AND-Splits und die Aufgaben A07, A11 und A17 die AND-Joins. Das zugehörige Workflow-Netz kann jetzt automatisch generiert werden und ist in Abbildung 14 dargestellt.

A01	Tragschicht säubern	A13	Offenporigen Asphalt walzen und verdichten
A02	Haftverbund herstellen	A14	Splittmastixasphalt endwalzen
A03	Asphaltbinderschicht herstellen	A15	Splittmastixasphalt hauptwalzen
A04	Einbau Gussasphalt heiß auf heiß	A16	Splitt für Gussasphalt anliefern
A05	Einbau Gussasphalt heiß auf kalt	A17	Deckschicht Splittmastixasphalt einbauen
A06	Unterlage für Offenporigen Asphalt abdichten	A18	Bindemittelumhüllenden Edelsplitt liefern
A07	Deckschicht Offenporigen Asphalt einbauen	A19	Splittmastixasphalt liefern
A08	Vorwalzen und einbringen von Splitt	A20	Vorbereitung Einbau Gussasphalt
A09	Deckschicht Offenporigen Asphalt walzen	A21	Vorbereitung Einbau Offenporiger Asphalt
A10	Deckschicht Offenporigen Asphalt säubern	A22	Vorbereitung Einbau Splittmastixasphalt
A11	Splitt auf Gussasphalt bringen	A23	Deckschicht säubern
A12	Splitt in Gussasphalt einwalzen		

Figure 11: Aufgaben für die Erstellung von Asphaltdeckschichten



- |   |  |
|---|--|
| ○ C01 Asphaltbinderschicht                        | ○ C11 Eingebauter Gussasphalt                      |
| ○ C02 Gesäuberte Tragschicht                      | ○ C12 Deckschicht Offenporigen Asphalt eingebaut   |
| ○ C03 Haftverbund hergestellt                     | ○ C13 Deckschicht Splittmastixasphalt eingebaut    |
| ○ C04 Einbau Gussasphalt vorbereitet              | ○ C14 Deckschicht Splittmastixasphalt vorgewalzt   |
| ○ C05 Einbau Offenporigen Asphalt vorbereitet     | ○ C15 Deckschicht Splittmastixasphalt hauptgewalzt |
| ○ C06 Einbau Splittmastixasphalt vorbereitet      | ○ C16 Fertiger Oberbau                             |
| ○ C07 Bindemittelumhüllenden Edelsplitt vorhanden | ○ C17 Splitt auf Gussasphalt gebracht              |
| ○ C08 Abgedichtete Unterlage                      | ○ C18 Deckschicht Offenporigen Asphalt gesäubert   |
| ○ C09 Splittmastixasphalt vorhanden               | ○ C19 Deckschicht Offenporigen Asphalt gewalzt     |
| ○ C10 Splitt vorhanden                            |  |

Figure 12: Voraussetzungen und Ergebnisse für die Erstellung von Asphaltdeckschichten

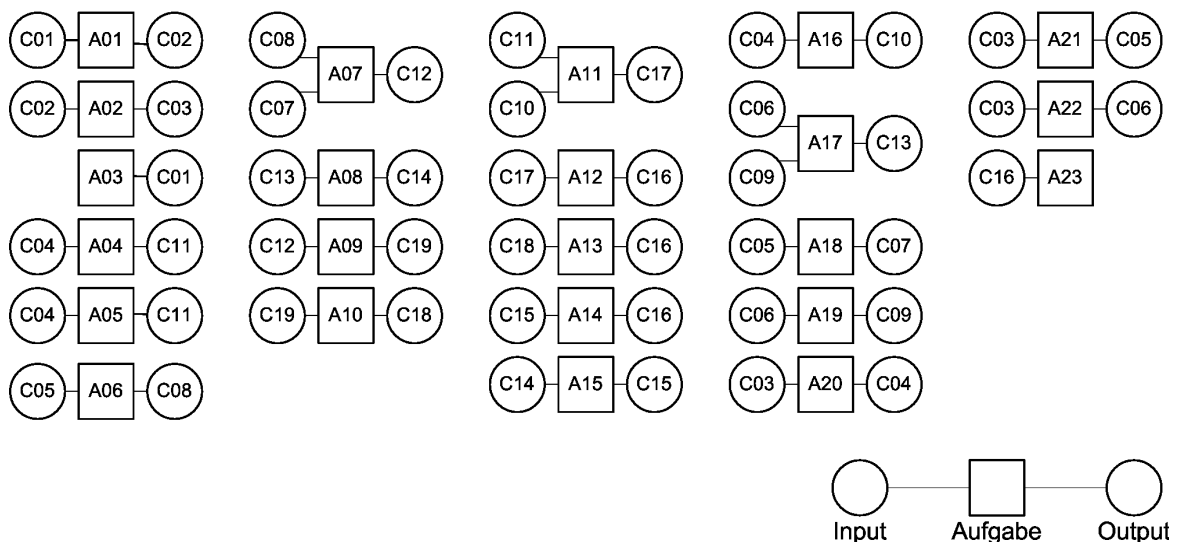


Figure 13: Zuordnung von Aufgaben und Bedingungen für die Erstellung von Asphaltdeckschichten

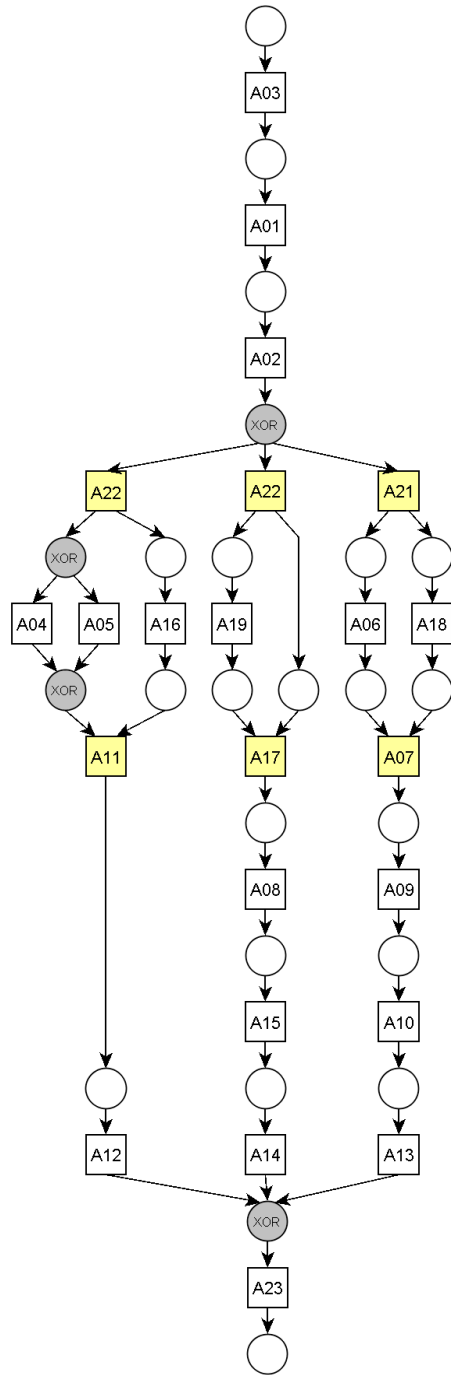


Figure 14: Generierter Workflow-Graph zur Erstellung von Asphaltdeckschichten

## 4 ZUSAMMENFASSUNG UND AUSBLICK

Mit Hilfe der vorgestellten Methode können Bauabläufe mit mehreren Varianten in Form von Workflow-Netzen automatisch erzeugt werden. Grundlage für die Generierung sind genau definierte Aufgaben mit allen notwendigen Voraussetzungen und deren Ergebnissen. Anhand eines Beispiels zur Erstellung von Asphaltdeckschichten wurde die entwickelte Vorgehensweise dargestellt. Die Erstellung, Anpassung und Pflege von umfangreichen Bauabläufen mit verschiedenen Varianten kann durch diesen Ansatz sinnvoll unterstützt werden.

Die Definition von Aufgaben und Bedingungen ist jedoch sehr zeitaufwendig. Im Rahmen weiterer Forschungsarbeiten soll ein fallbasiertes System zur Speicherung von durchgeführten Aufgaben und zugehörigen Voraussetzungen und Ergebnissen umgesetzt werden. Mit Hilfe dieses fallbasierten Systems sollen einmal festgelegte Aufgaben leichter wiederverwendet werden können. Ein weiterer zukünftiger Forschungsschwerpunkt wird in der Bewertung von Ausführungsvarianten liegen, um die Entscheidung für eine Variante zu unterstützen und transparent darzustellen.

## REFERENCES

- [1] Heinrich, T; Huhnt, W. 2003, Determination of Effects of Modifications during Planning Processes, 16. Internationalen Kolloquium über Anwendungen der Informatik und Mathematik in Architektur und Bauwesen , Weimar, 2003
- [2] Huhnt, W. 2005, Generating Sequences of Construction Tasks, Proceedings of 22nd of W78 Conference on Information Technology in Construction, Dresden, 2005.
- [3] Enge, F. 2005, Zustandsmodellierung als Grundlage für Ausführungsterminpläne, Forum Bauinformatik 2005, Cottbus, 2005
- [4] van der Aalst, W.M.P. 1998, The Application of Petri Nets to Workflow Management; The Journal of Circuits, Systems and Computers, Volume 8, Number 1, Pages 21-66, 1998.
- [5] Richter-von Hagen, C; Stucky, W 2004, Business-Process- und Workflow-Management, B. G. Teubner Verlag, Wiesbaden, 2004
- [6] Forschungsgesellschaft für Straßen- und Verkehrswesen: RStO-01 - Richtlinie für die Standardisierung des Oberbaus von Verkehrsflächen, FGSV-Verlag, Köln, 2001