H. Giese, K. Rosenthal (Hrsg.): Modellierung 2024 Satellite Events,
Digital Library, Gesellschaft für Informatik e.V., 2024 1

INTERFACE NETS TOOL: Modellierung von Netzkompositionen

Karl Ihlenfeldt, Daniel Moldt, Lukas Seifert und Laif-Oke Clasen[1]

Abstract: Die systematische Konstruktion von Systemen beinhaltet häufig die Erstellung eines Modells. Die Größe und Komplexität des Modells skaliert allerdings mit dem zu modellierenden System, was schnell zu unübersichtlichen und schwer wartbaren Modellen führen kann. Große oder komplexe Modelle werden deshalb häufig modularisiert. Die einzelnen Module werden dann zu dem Gesamtsystem zusammengeführt. Um diese Komposition des Gesamtsystems einheitlich und klar zu gestalten, hat Reisig das Konzept der Interface-Netze entwickelt, welches einen assoziativen Kompositionsoperator beinhaltet.
Interface-Netze sind eine spezielle Form von Petrinetzen. Obwohl es viele Werkzeuge gibt, welche die Modellierung mit Petrinetzen unterstützen, gibt es bislang noch kein Werkzeug, dass die Modellierung mit Interface-Netzen ermöglicht.
Um hierfür ein Werkzeug zu schaffen, wurde RENEW, ein existierendes Werkzeug für die Modellierung mit Petrinetzen, erweitert. RENEWS Plugin-Architektur unterstützt, neue Funktionalitäten zu implementieren und dabei auf bestehende Konzepten zurückzugreifen.
Das in diesem Beitrag vorgestellte Werkzeug, das INTERFACE NETS TOOL, ist eine Erweiterung RENEWS. Mit diesem Werkzeug ist es möglich, Interface-Netze zu modellieren und zu komponieren. Es wird auf die Funktionsweise des Werkzeuges eingegangen, sowie auf die Nutzung im wissenschaftlichen Kontext.
Das INTERFACE NETS TOOL unterstützt verschiedene Anwendungsmöglichkeiten. Neben der Komposition großer Systeme wird auch die Abstraktion von der Implementation ermöglicht, sodass die Schnittstellen der modellierten Module im Fokus stehen. Dies lässt sich auf die Modellierung von Multiagentensystemen und Web-Service basierten Systemen übertragen.

Keywords: Höhere Petrinetze; Interface-Netze; Renew; Modellierung; Komposition; Module

1 Einleitung

Große und komplexe Systeme bestehen häufig aus verschiedenen, miteinander kommunizierenden Einheiten. Sowohl die Konstruktion, insbesondere aber auch die Modellierung solcher Systeme erfordert die Betrachtung der einzelnen Komponenten und ihrer Schnittstellen. Diese Komponenten können unabhängig voneinander betrachtet oder gar ausgeführt werden, was gerade bei verteilten Systemen eine wichtige Rolle spielt. Unter anderem deshalb bieten sich Petrinetze als Modellierungstechnik für solche Systeme besonders an, da sie eine Modellierung von nebenläufigen Prozessen ermöglichen.

[1] Universität Hamburg, Algorithmen, Randomisierung und Theorie, Hamburg, Deutschland, http://paose.de
 https://doi.org/10.18420/modellierung2024-ws-021

Die anschließende Komposition der einzelnen Komponenten oder Netze zum Gesamtsystem erfordert allerdings eine einheitliche Modellierungstechnik. Hierfür hat Reisig die Modellierungstechnik der Interface-Netze entwickelt [Re09]. Mit dieser geht auch ein Kompositionsoperator einher, welcher eine assoziative Komposition der einzelnen Komponenten ermöglicht. Für die praktische Anwendung dieser Technik fehlt allerdings ein Software-basiertes Werkzeug. Ein solches Werkzeug wird in diesem Beitrag vorgestellt.

RENEW (siehe beispielsweise [KW99; Mo23b]) ist ein Open-Source Werkzeug der Universität Hamburg, welches die Erstellung, Simulation und Verifizierung von verschiedenen Arten von Petrinetzen ermöglicht. Durch die Plugin-basierte Architektur [Du09] wird eine Erweiterung der Anwendung um neue Funktionalitäten vereinfacht, zudem kann auf bereits implementierte Konzepte zurückgegriffen werden. Das in diesem Beitrag vorgestellte INTERFACE NETS TOOL stellt ein solches Plugin für RENEW dar.

Mit diesem neuen Werkzeug können neben der praktischen Modellierung komplexer Systeme auch neue Modellierungsansätze untersucht und getestet werden. Beispiele hierfür sind HERAKLIT INTERACTION DIAGRAMS [Mo23a] und eine Anwendung auf Web-Services in [Cl23], welche eine Kombination des PETRI NET-BASED, AGENT- AND ORGANIZATION-ORIENTED SOFTWARE ENGINEERING-Ansatzes [Ca10; Mo96] mit dem HERAKLIT-Ansatz [FR22] untersuchen.

In Abschnitt 2 werden die theoretischen Grundlagen aufgearbeitet, sowie die Umgebung erklärt, in welcher das Werkzeug entwickelt wurde. Abschnitt 3 beschreibt die Anforderungen an das Werkzeug, dessen Darstellung von Interface-Netzen, den angebotenen Funktionen und der Ausrichtung an RENEW. Abschnitt Abschnitt 4 zeigt ein Beispiel auf und geht auf die Stärken und Limitationen des Werkzeugs ein, bevor in Abschnitt 5 mit einem Ausblick auf zukünftige Anwendungen und Weiterentwicklungen abgeschlossen wird.

2 Grundlagen

Nachdem die theoretische Grundlage der Interface-Netze in Unterabschnitt 2.1 dargelegt wurde, befasst sich Unterabschnitt 2.2 mit der Umgebung, in welcher das Werkzeug entwickelt wurde. Unterabschnitt 2.3 beschreibt anschließend zwei Petrinetzformalismen, gefärbte Netze und Referenznetze, welche aufgrund ihrer Bedeutung für die Umgebung einen Einfluss auf die Ausrichtung des Werkzeuges hatten.

2.1 Interface-Netze

Reisig hat das Konzept der Interface-Netze im Laufe der Jahre weiterentwickelt und verallgemeinert [Re09; Re19]. In der ursprünglichen Veröffentlichung (siehe [Re09]) wurden Interface-Netze als eine spezielle Form elementarer Petrinetze (siehe beispielsweise [RE98]) definiert. Interface-Netze bestehen aus einem inneren Teil sowie einer linken und

einer rechten Schnittstelle (Interface). Plätze und Transitionen des Netzes können Teil einer (oder beider) Schnittstellen sein. In diesem Fall werden ihnen ein Label sowie ein Index zugewiesen, über welche sie eindeutig identifizierbar sind. Dies spielt eine elementare Rolle für die Komposition der Netze.

Grafisch werden Interface-Netze in [Re09] als Boxen dargestellt; der linke und rechte Rand der Box werden als Interfaces des Netzes verstanden. Elemente im Interface liegen auf dem jeweiligen Rand, ihr Label wird neben ihnen notiert. Der Index des Elementes wird implizit durch dessen Position im Interface dargestellt. In [Re09] haben höherliegende Elemente einen höheren Index, welcher bei den niedriger gelegenen Elementen verringert wird. In späteren Veröffentlichungen, wie zum Beispiel [Re19], wurde diese Reihenfolge umgedreht. Das in diesem Beitrag vorgestellte Werkzeug verwendet die neuere Darstellung, bei der höherliegende Elemente einen niedrigeren Index haben.

Zwei Interface-Netze können zu einem neuen Interface-Netz komponiert werden. Hierbei wird das linke Interface des einen mit dem rechten Interface des anderen Netzes verschmolzen. Haben zwei Elemente des gleichen Typs dasselbe Label und denselben Index, werden sie zu einem einzigen Element verschmolzen. Kanten, die mit den ursprünglichen Elementen verbunden waren, werden stattdessen mit dem neuen Element verbunden, sodass die Netzstruktur erhalten bleibt. Elemente ohne passendes Gegenstück werden dem jeweils linken oder rechten Interface des neuen Netzes unten hinzugefügt, abhängig von ihrem vorherigen Interface. Ein Beispiel findet sich in Abbildung 1. Wie von Reisig gezeigt, ist dieser Kompositionsoperator (Notation: $\bullet$) assoziativ [Re19], aber im Allgemeinen nicht kommutativ.

2.2 Renew

RENEW (The Reference Net Workshop) wird als Open-Source Software an der Universität Hamburg seit vielen Jahren entwickelt (siehe beispielsweise [Ku23]). RENEW ist vor allem ein Editor und Simulator für eine Vielzahl von Petrinetzformalismen, wie zum Beispiel P/T-Netze, gefärbte Petrinetze (siehe zum Beispiel Unterabschnitt 2.3 oder [Je97]) oder auch Referenznetze (siehe Unterabschnitt 2.3 oder [Ku02]). Der Referenznetzformalismus ist dabei von zentraler Bedeutung in RENEWs Historie.

Die Anwendung erlaubt dem Nutzer, beliebe Anordnungen von Plätzen, Transitionen und Kanten in ein *Drawing* zu zeichnen und als Datei abzuspeichern oder zu simulieren. Es bleibt dabei der Interpretation des Nutzers überlassen, ob eine Zeichnung ein nicht-zusammenhängendes Petrinetz darstellt, oder ob mehrere Petrinetze in einem Drawing enthalten sind. Dies wird in Unterabschnitt 3.2 nochmal aufgegriffen.

RENEWs modularisierte Plugin-Architektur (siehe [Mo23b]) erlaubt es dem Nutzer, beim Start der Anwendung nach Belieben bestimmte Plugins zu laden. Jedes Plugin kann dabei als eine Sammlung von Funktionen und Regeln verstanden werden, welche dem Nutzer weitere

Modellierungsmöglichkeiten eröffnen. Das hier vorgestellte Werkzeug, das INTERFACE NETS TOOL, ist ein solches Plugin.

2.3 Gefärbte Netze und Referenznetze

Referenznetze (siehe [Ku02]) sind eine Erweiterung der gefärbten Netze. Sie sind einer der zentralen Aspekte RENEWS und haben auch einen Einfluss auf die Ausrichtung des INTERFACE NETS TOOLS. In diesem Abschnitt werden deshalb die wichtigsten Faktoren hervorgehoben.

Gefärbte Petrinetze (siehe zum Beispiel [Je97]) sind P/T-Netze, bei denen die Marken Werte (auch: Farben) annehmen können. Die Kanten, Transitionen und Schaltregeln der gefärbten Netze berücksichtigen dies. Kanten und Transitionen können mit Anschriften versehen werden, um sicherzustellen, dass eine Transition nur bei dem Vorhandensein bestimmter Werte(-Typen) schalten kann oder den eingehenden Wert auf eine bestimmte Weise verändert. Beispiele hierfür sind sogenannte *guards* und *Variablen*. Variablen sind Anschriften für sowohl Kanten als auch Transitionen, welche als Platzhalter für eine spätere Marke eines bestimmten Wertes dienen. Sie gelten nur im Bereich der direkt mit der Transition verbundenen Elemente (ihrer *Lokalität*). Guards sind Anschriften einer Transition, welche eine Bedingung darstellen. Diese muss eingehalten werden, damit die jeweilige Transition feuern kann. Guard-Anschriften können auch Variablen nutzen.

Referenznetze sind Java-basiert und können als Java-Objekt angesehen werden. Weiter können die Marken Referenzen auf andere Netze darstellen. Auf diese Weise verschachtelte Netze können in der Referenznetzsemantik miteinander über *synchrone Kanäle* [CH92] kommunizieren. Ein synchroner Kanal ähnelt einem Methoden-Aufruf und stellt eine weitere Art einer Transitionsanschrift dar. Er besteht aus einem Uplink und einem Downlink. Der Uplink kann als die Signatur einer Methode verstanden werden; er definiert den Namen und mögliche Parameter des Kanals. Der Downlink kann, wie ein Methodenaufruf in objektorientierten Programmiersprachen, anhand einer Referenz auf ein Netz dessen Uplinks aufrufen. Die aufrufende Transition synchronisiert sich mit der aufgerufenen Transition und feuert mit dieser zusammen. Es ist wichtig zu beachten, dass Transitionen maximal einen Uplink definieren können, aber beliebig viele Downlinks aufrufen können.

3 INTERFACE NETS TOOL

Das INTERFACE NETS TOOL ist eine Plugin RENEWS, welche es dem Nutzer ermöglicht, Interface-Netze zu erstellen, editieren, komponieren und simulieren. Hierbei greift es auf verschiedene, bereits existierende Funktionen RENEWS zurück. Dem Nutzer werden neue Elemente angeboten, wie die Interfaces, oder auch das Interface-Netz, welche durch das neue Plugin komponiert werden können. Des Weiteren wird aus technischen Gründen ein neuer Drawing-Typ eingeführt.

Innerhalb eines Drawings kann die Komposition zweier Netze durch den Nutzer per Drag & Drop angestoßen werden. Die Komposition kann auch über mehrere Drawings hinweg erfolgen, indem ein spezielles Bedienfenster, die *Composition-Gui*, benutzt wird. Die Composition-Gui kann außerdem dafür verwendet werden, mehrere Interface-Netze über eine Formeleingabe automatisch zu komponieren.

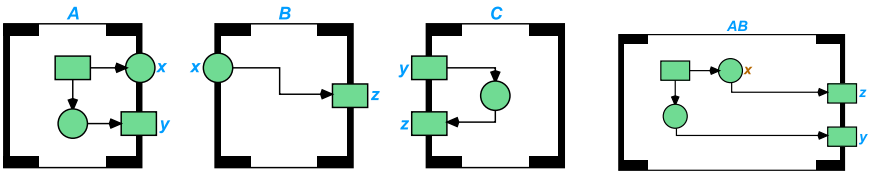
In Unterabschnitt 3.1 werden die Anforderungen an das Werkzeug vorgestellt, bevor Unterabschnitt 3.2 die Darstellung von Interface-Netzen mit dem Werkzeug präsentiert. In Unterabschnitt 3.3 wird eine weitere Operation des Werkzeuges dargestellt. Unterabschnitt 3.4 stellt die Composition-Gui vor, bevor in Unterabschnitt 3.5 die Handhabung von Referenznetzen durch das Werkzeug beschrieben wird.

3.1 Anforderungen

Neben der korrekten Darstellung und Komposition von Interface-Netzen werden verschiedene Anforderungen an das INTERFACE NETS TOOL gestellt.
Es soll ein schnelles, einfaches Entwerfen von simplen Systemen ermöglicht werden. Dies könnte beispielsweise für die universitäre Lehre genutzt werden, für Brainstorming-Phasen oder oberflächliche Diskussionen über grundlegende Modellierungsentscheidungen, oder auch dem einfachen Testen von Konzepten.
Das Erstellen und die Analyse großer, komplexer Modelle soll vereinfacht werden. Dafür ist eine Modularisierung der einzelnen Komponenten auf mehrere Drawings hilfreich.
Das Werkzeug soll von anderen Plugins genutzt werden können. Hierzu gehört zum einen die Generierung von Interface-Netzen anhand von AGENT INTERACTION PROTOCOL DIAGRAMS (siehe [CM05]) durch das DIAGRAM TOOL (siehe [Se24]) und zum anderen die Verwendung als Grundlage für ein mögliches HERAKLIT-Modellierungswerkzeug (siehe [FR24], oder Abschnitt 5).
Das Werkzeug soll eine Verwendung von Referenznetzen in Verbindung mit Interface-Netzen unterstützen.

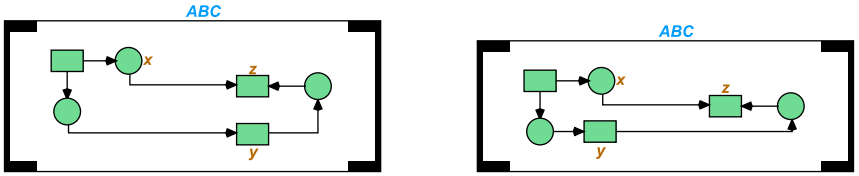
3.2 Darstellung von Interface-Netzen

Die flexible Aufteilung von Petrinetzen auf Drawings, welche RENEW bereitstellt, erfordert ein neues Element, die *Interface-Box*. Innerhalb eines Drawings stellt die Interface-Box ein vollständiges Interface-Netz dar, welches somit klar von anderen Interface-Netzen unterschieden werden kann. Abbildung 1a zeigt die Darstellung von drei Interface-Netzen, *A*, *B* und *C*, als Interface-Boxen. Jede Box enthält genau zwei Interfaces, welche als schwarze, eckige Klammern visualisiert werden. Netz *A* hat zwei Elemente im rechten Interface, welche auch grafisch dort verortet sind. Das Label der Elemente, *x* für den Platz und *y* für die Transition, wird in blauer Farbe neben den jeweiligen Elementen angezeigt. Der jeweilige Index wird implizit durch die Position im Interface dargestellt. Über der



(a) Drei Interface-Netze A, B und C

(b) Das Ergebnis der Komposition von $A \cdot B$



(c) Das Ergebnis der Komposition von $(A \cdot B) \cdot C$

(d) Das Ergebnis der Komposition von $A \cdot (B \cdot C)$

Abb. 1: Darstellung dreier Interface-Netze und ihrer Komposition mit dem INTERFACE NETS TOOL

Interface-Box befindet sich in blauer Schrift der Name des Interface-Netzes, welcher für die Komposition durch eine Formeleingabe genutzt wird (siehe Unterabschnitt 3.4).

Die Interface-Boxen können per Drag & Drop miteinander komponiert werden. Hierbei werden die anliegenden Interfaces der jeweiligen Netze zusammengeführt. Elemente, die dabei vereinigt werden, werden durch ein neues Element im Inneren des neuen Netzes ersetzt. Ein Beispiel hierfür wird in Abbildung 1b gezeigt. Durch die Komposition der Netze $A \cdot B$ wurden die Plätze mit dem Label x verschmolzen. Das neue Element befindet sich nicht mehr im Interface, was durch einen Farbwechsel des Labels dargestellt wird. Auch der Name des komponierten Netzes wurde angepasst, indem die Namen der komponierten Netze konkateniert wurde.

Der von Reisig definierte Kompositionsoperator ist assoziativ. Unabhängig von der Ausführungsreihenfolge resultiert die Komposition von mehreren Interface-Netzen immer in der selben Netzstruktur. Dies gilt auch für die Implementierung des Operators in dem hier vorgestellten Werkzeug. Es ist allerdings zu bemerken, dass sich die grafische Anordnung der Elemente durch die Kompositionsreihenfolge verändern kann. Dies wird in den Abbildungen 1c und 1d verdeutlicht. Die Netzstruktur beider Netze ist identisch, allerdings wurde in Abbildung 1c die Komposition $A \cdot B$ zuerst ausgeführt, wodurch die Transition mit dem Label y in das rechte Interface des resultierenden Netzes verschoben wurde. In Abbildung 1d konnte diese Verschiebung nicht stattfinden, wodurch sich die Transition in einer anderen Position befindet.

3.3 Vereinigungsoperator

Zusätzlich zum Kompositionsoperator wurde ein Vereinigungsoperator implementiert. Mit diesem kann ein Interface-Netz generiert werden, welches aus zwei disjunkten Subnetzen besteht. Anstatt zwei Interface-Netze miteinander zu komponieren und passend gelabelte Elemente miteinander zu verschmelzen, werden die Interface-Netze mit dieser Operation nicht auf zueinander passende Elemente überprüft; stattdessen werden die jeweils linken (und rechten) Interfaces der beiden Netze vereint.

Das DIAGRAM TOOL nutzt diese Operation, um anhand von AGENT INTERACTION PROTOCOL DIAGRAMS (eine spezielle Form von Sequenzdiagrammen, siehe [Ca10]) verschiedene Interface-Netze zu vereinigen. Damit können Abläufe in der Form eines Interface-Netzes dargestellt werden. Im Kontext vom DIAGRAM TOOL repräsentieren die Schnittstellenelemente den Nachrichtenaustausch zwischen zwei Agenten. Die Vereinigungsoperation wurde entwickelt, weil die Komposition zweier Interface-Netze in einigen Fällen zu falschen Modellen führen kann: Wenn ein Agent eine Nachricht sowohl empfangen als auch versenden soll, die dazugehörigen Schnittstellenelemente aber das selbe Label haben, würden diese Elemente fälschlicherweise miteinander verschmolzen und dem inneren Teil des Interface-Netzes zugeordnet werden.

3.4 Composition-Gui

Die Composition-Gui wurde entwickelt, um eine Komposition von Interface-Netzen über mehrere Drawings hinweg zu ermöglichen. Des Weiteren soll die Komposition mehrerer Interface-Netze vereinfacht und automatisiert werden. Abbildung 2 zeigt RENEW mit vier geöffneten Interface-Drawings und der Composition-Gui im Vordergrund. Die Drawings A, B und C aus Abbildung 1 sind hier als eigene Zeichnungen geöffnet. Das vierte Drawing stellt das Ergebnis der Komposition von $(A \cdot B) \cdot C$, vereinigt mit einer Kopie von sich selbst, dar; ausgedrückt durch die Formel $A^*B^*C + A^*B^*C$.

Des Weiteren wird das Formeleingabefeld für die automatische Komposition angeboten. Die Formeln sind syntaktisch ähnlich zu Formeln für die Multiplikation und Addition von Zahlen, nur dass hier Label statt Ziffern verwendet werden. Sie können beliebig verschachtelte Klammern enthalten und der Kompositionsoperator (*) hat Vorrang gegenüber dem Vereinigungsoperator (+). Ein Beispiel kann in Abbildung 2 betrachtet werden.

3.5 Kompatibilität mit Referenznetzen

Eine von RENEWS Besonderheiten ist der Referenznetzformalismus (siehe Unterabschnitt 2.3). Um eine möglichst vielseitige Verwendung des Werkzeuges zu ermöglichen, wurde der Versuch unternommen, diesen mit dem Konzept der Interface-Netz zu verbinden.

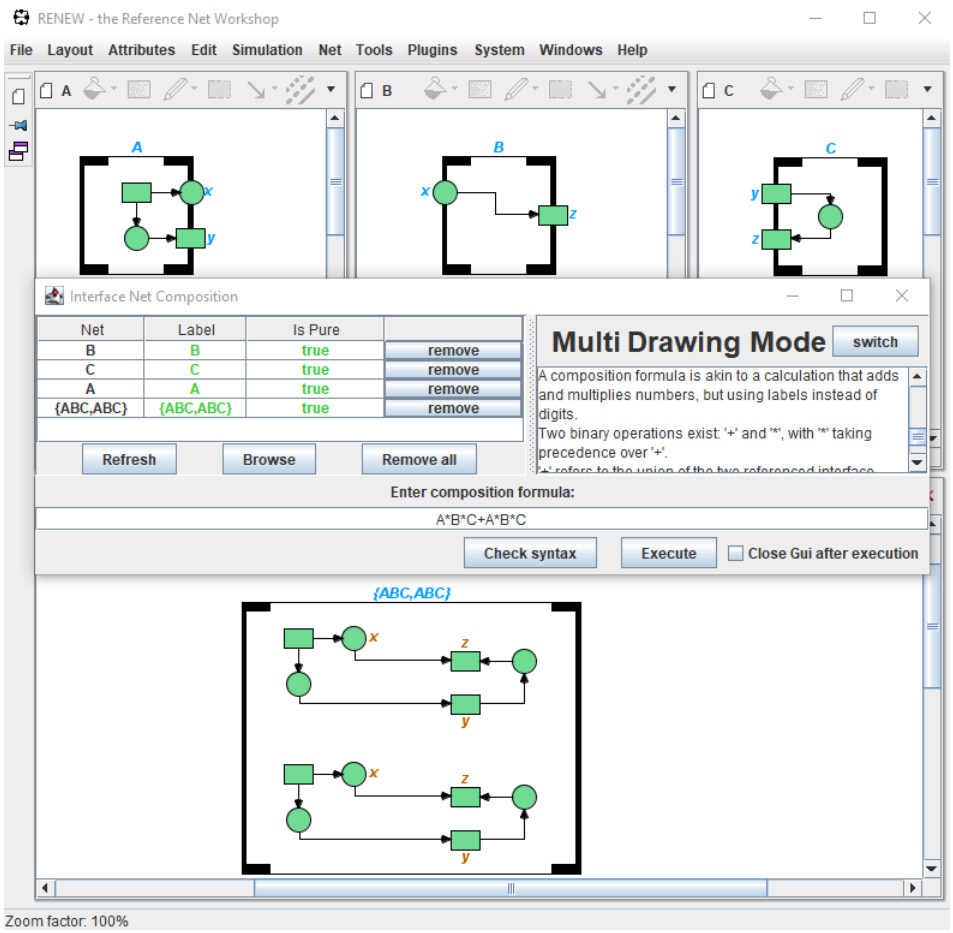
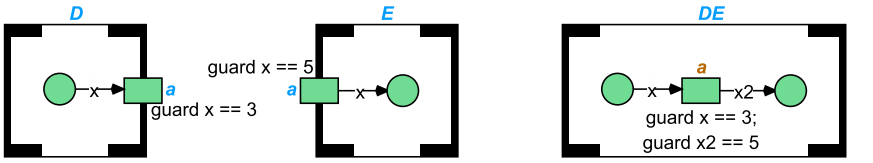


Abb. 2: RENEW mit 4 Interface-Netzen und der Composition-Gui

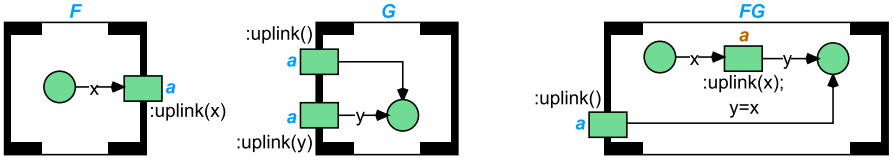
Hierbei wurden zwei Fälle von besonderen Transitionsanschriften betrachtet: Anschriften mit Variablen und Uplinks. Anschriften ohne Variablen, sowie Downlinks, können vereinigt werden, ohne das Netzverhalten zu beeinflussen.

Variablen gelten für die Lokalität einer Transition (siehe Unterabschnitt 2.3). Da bei der Komposition die Lokalitäten zweier zu verschmelzender Transitionen vereinigt werden, prüft das Werkzeug die Transitionsanschriften auf mögliche Variablen mit Namenskonflikten. Bestehende Namenskonflikte werden automatisch durch eine passende Umbenennung aufgelöst. Ein Beispiel hierfür wird in Abbildung 3 gezeigt. Beide Transitionen nutzen eine Variable x in ihrer Lokalität. Die Variable der rechten Transition wurde sowohl in



(a) Interface-Netze mit sich gegenseitig ausschließenden guards (b) Das Ergebnis der Komposition von $D \cdot E$

Abb. 3: Auflösung von widersprüchlichen guards durch das INTERFACE NETS TOOL



(a) Interface-Netze mit verschiedenen Uplinks (b) Das Ergebnis der Komposition von $F \cdot G$

Abb. 4: Handhabung von synchronen Kanälen durch das INTERFACE NETS TOOL

der Transitionsanschrift, als auch in der Kantenanschrift umbenannt, um den Konflikt aufzulösen.

Des Weiteren werden Uplink-Definitionen wie eine Erweiterung des Labels betrachtet. Transitionen mit dem selben Label werden nur verschmolzen, wenn maximal eine der Transitionen einen Uplink definiert oder aber beide Transitionen einen Uplink mit dem selben Namen und der selben Anzahl Parameter definieren. In beiden Fällen behält die neue Transition die Uplink-Anschrift. Im zweiten Fall werden außerdem die Parameter aufeinander abgebildet, um das Netzverhalten nicht zu verändern. Abbildung 4 zeigt hierfür ein Beispiel.

4 Evaluation

Die Funktionalitäten des Werkzeuges wurden bislang hauptsächlich im Zuge der Entwicklung und Forschung getestet. Eine Evaluierung anhand von Praxiserfahrungen ist zum jetzigen Zeitpunkt stark eingeschränkt.

Das vorgestellte Werkzeug ermöglicht es, Interface-Netze zu erstellen, simulieren und komponieren. Hierfür wurde die theoretische Grundlage (siehe beispielsweise [Re09; Re19]) als Maßstab genommen. Die Interfaces wurden für die einfachere Nutzung als schwarze Klammern stärker hervorgehoben und die Farbe der Label wurde als Indikator für Elemente im oder außerhalb des Interfaces genutzt. Wie auch in [Re09] wurden die Indizes der

Elemente implizit über die Position im Interface dargestellt, wobei sich die Ordnung an [Re19] orientiert.

In [Mo23a] und [Cl23] wurde untersucht, wie die Modellierungsansätze PAOSE [Ca10; Mo96] (PETRI NET-BASED, AGENT- AND ORGANIZATION-ORIENTED SOFTWARE ENGINEERING) und HERAKLIT [FR22; FR24] miteinander in Beziehung gesetzt werden können. Hierfür wurden das hier vorgestellte INTERFACE NETS TOOL und das DIAGRAM TOOL (siehe [Se23; Se24]) verwendet, um das Zusammenspiel der beiden Modellierungskonzepte zu testen und zu verfeinern. Eine vereinfachte Version des INTERFACE NETS TOOLS wurde erfolgreich verwendet, um die Zusammenführung der Konzepte zu testen und zu visualisieren.

Auch das einfache Entwerfen von Interface-Netzen wird unterstützt. Das Erstellen der Interface-Boxen (siehe Unterabschnitt 3.2) sowie deren Komposition per Drag & Drop vereinfachen die Nutzung und erfordern keine aufwendige Einrichtung. Die Komposition mehrerer Netze kann durch eine Formeleingabe automatisiert werden.

Die Modellierung komplexer Systeme wird durch eine Modularisierung der Komponenten auf mehrere Drawings unterstützt. Mit der Composition-Gui ist es möglich, die Komponenten über mehrere Drawings hinweg zu komponieren. Dadurch wird ein hoher Grad an Modularisierung unterstützt, bei dem jede Komponente in einer eigenen Datei verwaltet wird. Große, komplexe Systeme können so in ihre Bestandteile zerlegt werden, um diese anschließend mit einer Formeleingabe zum Gesamtsystem zusammenzuführen.

Das INTERFACE NETS TOOL kann von anderen Plugins genutzt werden. Ein Beispiel hierfür ist das DIAGRAM TOOL [Se23; Se24]. Das Werkzeug erlaubt es gefärbte Petrinetze und Referenznetze mit den Interface-Netzen zu verbinden. Hierfür gibt es keine formale Grundlage und die Auswirkung der implementierten Regeln auf den Kompositionsoperator sowie die Modellierungskonzepte von HERAKLIT ist noch nicht formalisiert worden.

Mit dem vorgestellten Werkzeug ist auch eine Simulation der Netze möglich. Eine Komposition während der Simulation ist allerdings nicht möglich, da der Zustand der Netze im jeweiligen Simulationsschritt von diesem Werkzeug nicht in Betracht gezogen werden kann.

Da das INTERFACE NETS TOOL bisher nicht veröffentlicht wurde, erfolgte keine externe Evaluation. Intern wurde es erfolgreich für den Entwurf von Modulen und Netzen in kleineren Aufgaben eingesetzt. Eine Erprobung in einem größeren Projekt ist für ein Lehrprojekt im nächsten Semester vorgesehen.

5 Ausblick

Aufgrund der chronologischen und thematischen Verbindung zwischen den Modellierungsansätzen der Interface-Netze und HERAKLIT liegt es nahe, das INTERFACE NETS TOOL

zu nutzen, um ein HERAKLIT Werkzeug zu entwickeln. Versuche in diese Richtung werden momentan unternommen.

Die Modellierung der statischen Beziehungen von Modulen und die Einbettung in einen gesamten Systementwicklungsansatz erfordert weitere Modellierungstechniken. Diese sollen, wie beim PAOSE Ansatz und dem HERAKLIT Ansatz, Petrinetz basiert sein, damit eine konsistente Abstimmung der Modellierungstechniken gelingt.

Literatur

[Ca10] Cabac, L.: Modeling Petri Net-Based Multi-Agent Applications, Dissertation, Vogt-Kölln Str. 30, D-22527 Hamburg: Universität Hamburg, Department Informatik, Apr. 2010, URL: https://ediss.sub.uni-hamburg.de/handle/ediss/3691.

[CH92] Christensen, S.; Hansen, N. D.: Coloured Petri Nets Extended with Channels for Synchronous Communication, Techn. Ber. DAIMI PB-390, Aarhus University, 1992.

[Cl23] Clasen, L.-O.; Moldt, D.; Hansson, M.; Ihlenfeldt, K.; Seifert, L.: Associative Composition of Web Service Invocation Sequences Based on Agent-Interaction-Diagrams. In (Michael Köhler-Bußmeier Wolfgang Renz, J. S., Hrsg.): Intelligent Distributed Computing XVI: 16th International Symposium on Intelligent Distributed Computing, IDC 2023. A Springer book series Studies in Computational Intelligence, Springer Nature Switzerland AG, Cham, Switzerland, 2023.

[CM05] Cabac, L.; Moldt, D.: Formal Semantics for AUML Agent Interaction Protocol Diagrams. In: Agent-Oriented Software Engineering V: 5th International Workshop, AOSE 2004, New York, NY, USA, July 19, 2004. Revised Selected Papers. Bd. 3382. Lecture Notes in Computer Science, Springer-Verlag, S. 47–61, Jan. 2005, URL: http://dx.doi.org/10.1007/978-3-540-30578-1_4.

[Du09] Duvigneau, M.: Konzeptionelle Modellierung von Plugin-Systemen mit Petrinetzen, https://ediss.sub.uni-hamburg.de/handle/ediss/3023, Dissertation, Vogt-Kölln Str. 30, D-22527 Hamburg: Universität Hamburg, Department Informatik, Okt. 2009, URL: https://ediss.sub.uni-hamburg.de/handle/ediss/3023.

[FR22] Fettke, P.; Reisig, W.: Modularization, Composition, and Hierarchization of Petri Nets with Heraklit, 2022, arXiv: 2202.01830 [cs.SE].

[FR24] Fettke, P.; Reisig, W.: The HERAKLIT Monograph — Understanding Computer-Integrated Systems — How to model the digital world with HERAKLIT; Preprint / private communication; to be published, 2024, URL: http://www.heraklit.org.

[Je97] Jensen, K.: A brief introduction to coloured Petri Nets. In (Brinksma, E., Hrsg.): Tools and Algorithms for the Construction and Analysis of Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 203–208, 1997, ISBN: 978-3-540-68519-7.

[Ku02] Kummer, O.: Referenznetze. Logos Verlag, Berlin, 2002, ISBN: 978-3-8325-0035-1.

[Ku23] Kummer, O.; Wienberg, F.; Duvigneau, M.; Cabac, L.; Haustermann, M.; Mosteller, D.: Renew – The Reference Net Workshop, Release 4.1, Feb. 2023, URL: http://www.renew.de/.

[KW99] Kummer, O.; Wienberg, F.: Renew – The Reference Net Workshop, Available at: http://www.renew.de/, Release 1.0, März 1999, URL: http://www.renew.de/.

[Mo23a] Moldt, D.; Hansson, M.; Seifert, L.; Ihlenfeldt, K.; Clasen, L.; Ehlers, K.; Feldmann, M.: Enriching Heraklit Modules by Agent Interaction Diagrams. In (Gomes, L.; Lorenz, R., Hrsg.): Application and Theory of Petri Nets and Concurrency - 44th International Conference, PETRI NETS 2023, Lisbon, Portugal, June 25-30, 2023, Proceedings. Bd. 13929. Lecture Notes in Computer Science, Springer Nature Switzerland AG, Cham, Switzerland, S. 440–463, 2023, URL: https://doi.org/10.1007/978-3-031-33620-1_23.

[Mo23b] Moldt, D.; Johnsen, J.; Streckenbach, R.; Clasen, L.; Haustermann, M.; Heinze, A.; Hansson, M.; Feldmann, M.; Ihlenfeldt, K.: RENEW: Modularized Architecture and New Features. In (Gomes, L.; Lorenz, R., Hrsg.): Application and Theory of Petri Nets and Concurrency - 44th International Conference, PETRI NETS 2023, Lisbon, Portugal, June 25-30, 2023, Proceedings. Bd. 13929. Lecture Notes in Computer Science, Springer Nature Switzerland AG, Cham, Switzerland, S. 217–228, 2023, URL: https://doi.org/10.1007/978-3-031-33620-1_12.

[Mo96] Moldt, D.: Höhere Petrinetze als Grundlage für Systemspezifikationen, Dissertation, Vogt-Kölln Str. 30, D-22527 Hamburg: Universität Hamburg, Fachbereich Informatik, Aug. 1996.

[Re09] Reisig, W.: Simple Composition of Nets. In (Franceschinis, G.; Wolf, K., Hrsg.): Applications and Theory of Petri Nets, 30th International Conference, PETRI NETS 2009, Paris, France, June 22-26, 2009. Proceedings. Bd. 5606. Lecture Notes in Computer Science, Springer-Verlag, S. 23–42, 2009, URL: https://doi.org/10.1007/978-3-642-02424-5%5C_4.

[Re19] Reisig, W.: Associative composition of components with double-sided interfaces. Acta Informatica 56/3, S. 229–253, 2019.

[RE98] Rozenberg, G.; Engelfriet, J.: Elementary net systems. In (Reisig, W.; Rozenberg, G., Hrsg.): Lectures on Petri Nets I: Basic Models: Advances in Petri Nets. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 12–121, 1998, ISBN: 978-3-540-49442-3, URL: https://doi.org/10.1007/3-540-65306-6_14.

[Se23] Seifert, L.: Erweiterung der Modelltransformationen von Agent Interaction Protocols in Renew durch Generalisierung des Diagram Plugins, Bachelorarbeit, Vogt-Kölln Str. 30, D-22527 Hamburg: Universität Hamburg, Fachbereich Informatik, Sep. 2023.

[Se24] Seifert, L.; Moldt, D.; Ihlenfeldt, K.; Clasen, L.-O.: DiagramTool, Tool-Paper für die Modellierungskonferenz 2024, 2024.