Софийски университет „Св. Климент Охридски“
Факултет по математика и информатика
Катедра „Компютърна информатика“

ас. Димитър Георгиев Димитров

Програмен аспект на теорията на обобщените мрежи – оптимизация на алгоритми за изпълнение, оператори за модификация на модели и приложения

АВТОРЕФЕРАТ

НА ДИСЕРТАЦИЯ

за присъждане на образователна и научна степен „Доктор“
по професионално направление
4.6 „Информатика и компютърни науки“
научна специалност: 01.01.12 „Информатика“

Научен ръководител:
чл.-кор. проф. дмн дтн Красимир Атанасов,
ИБФБМИ-БАН

София, 2013

Съдържание

Въведение

Обобщените мрежи (ОМ) представляват разширение на мрежите на Петри (МП). ОМ са инструмент за моделиране и оптимизация на паралелни и конкурентни процеси в сложни системи.

Актуалност на проблема

През годините са създадени стотици ОМ модели на процеси от различни области от науката и промишлеността. По отношение на медицината например са създадени над 800 ОМ модела на процеси на поставяне на диагнози на различни заболявания и описание на функционирането на цели системи в човешкия организъм. В областта на изкуствения интелект са конструирани десетки ОМ модели на експертни системи и бази от данни, невронни мрежи, системи за вземане на решения и оптимизация, разпознаване на образи и реч, процеси в машинното обучение, генетични алгоритми, гъвкави автоматизирани производствени и роботизирани системи и други. За всяка от споменатите групи обекти е показано, че съществува ОМ, описваща функционирането и резултатите от работата им, т.е. апаратът на ОМ успешно може да послужи за моделирането на всички тези обекти, въпреки разнородната природа на техните математически основи и програмна реализация. ОМ са намерили приложение още в биотехнологиите, софтуерното инженерство (например връзка между UML и ОМ), електронното обучение, химическата индустрия (Химко-Враца), петролната индустрия (Нефтохим-Бургас), транспорта, телекомуникациите, мениджмънта, икономиката, администрацията, физиката, астрономията и други.

В теорията на ОМ са обособени седем аспекта: алгебричен, топологичен, логически, операторен, програмен, методологичен и дидактически. Предмет на програмния аспект е създаването и развиването на инструментариум за машинно представяне и симулация на ОМ модели. Обект на настоящия дисертационен труд са именно изследвания на този аспект: обзор на съществуващи решения, оптимизация на алгоритми, разширяване на възможностите на програмното осигуряване. Процесът на изследване на програмния аспект е тясно свързан с раз-

ширение и подобряване на елементи от други аспекти.

Моделиращите качества на ОМ са обект на много теоретични изследвания, но за съжаление част от тях не са придружени с програмна реализация. Една цялостна софтуерна платформа за симулация с ОМ ще демонстрира на практика възможностите на това средство за моделиране и ще даде на специалистите мощен инструмент, чрез който да се възползват от пълните възможности на ОМ. Един добър програмен пакет е необходимо да притежава поне следните функционалности, за да представлява интегрирана среда за моделиране:

- визуално конструиране и редактиране на ОМ модели;
- симулация на ОМ модели;
- инспектиране на състоянието на ОМ по време на симулация;
- удобен потребителски интерфейс;
- интеграция със съществуващи среди за числови и символни пресмятания (като MATLAB, Maple, Mathematica).

През годините са разработвани различни софтуерни решения за моделиране с обобщени мрежи. При повечето от тях има известна обща функционалност, която обаче в почти всички случаи е реализирана независимо от останалите, започвайки отначало. В резултат почти липсва съвместимост и приемственост между различните софтуерни приложения. Освен това много софтуерни приложения за ОМ са създавани с цел симулацията на конкретен модел и не предвиждат разширяемост.

Добър резултат в тази насока е започването на ново поколение софтуер в ИБФБМИ на БАН и Техническия университет в Дрезден. Разработен е негов основен компонент – GNTicker, който въвежда набор от формати и протоколи, например единен разширяем XML формат за описание на ОМ модели. Логична следваща стъпка е създаването на интегрирана среда за моделиране чрез ОМ, която дава удобен достъп до голяма част от новоизброените възможности. Очаква се тя да улесни и създаването на нови софтуерни компоненти, свързани с моделирането с обобщени мрежи, благодарение на предвидените възможности за разширяемост и повторно използване на вече реализирана функционалност.

Визуална среда с удобен потребителски интерфейс, чрез която се

извършва целият процес по моделиране, ще популяризира обобщените мрежи сред експерти от различни приложни области.

Най-добре предимствата на една интегрирана среда за моделиране се виждат при използването ѝ с примерни ОМ модели – както такива с приложен характер, така и такива с теоретичен характер.

По същия начин, по който едно от най-големите теоретични предимства на ОМ е способността им да представят голямо разнообразие от други моделни средства, така и едно от големите практически предимства на разработения към този труд софтуер е възможността да се симулират модели, разработени с такива средства. Към момента са разработени ОМ модели, описващи например различните мрежи на Петри. Особена роля при представянето на детерминистични последователни процеси, комуникиращи с неограничени канали от тип „първи влязъл, първи излязъл“, имат мрежите на Кан. Въпросът за тяхната симулация и моделиращите им възможности са интересни за изследване, например в областта на обработката на цифрови сигнали.

Цел и задачи

Основната цел на настоящия дисертационен труд е проектирането и разработката на интегрирана среда за моделиране с обобщени мрежи. Средата предлага възможност за оптимизация и трансформиране с цел ефективно изпълнение на разработените модели.

За постигането на поставената цел се дефинират следните задачи:

- Да се анализират и изследват съществуващите софтуерни средства за работа с ОМ, като се опишат техните характеристики, предимства, недостатъци, възможност за използване съвместно с настоящия софтуерен пакет.
- Да се модифицират и реализират алгоритми за движение на ядрата в ОМ и някои техни разширения с цел по-ефективно изпълнение.
- Да се модифицират и реализират трансформационни оператори за ОМ с цел постигане на модулност и повишаване на гъвкавостта на моделиране.

- Да се разгледат възможностите за построяване на универсална ОМ, която симулира работата на произволна мрежа на Кан.
- Да се проектира и реализира интегрирана среда за моделиране с ОМ.
- Да се валидира разработената среда за моделиране и интеграцията с външна система на базата на ОМ в областта на биотехнологичните процеси.

Структура

В първа глава се прави представяне на обобщените мрежи като средство за моделиране и се дава сравнение с мрежите на Петри. Разглеждат се последователно формалната дефиниция на ОМ, алгоритъмът за тяхното функциониране и методологията за изграждане на ОМ.

Във втора глава се прави аналитичен обзор на съществуващите софтуерни решения за работа с ОМ. Описани са характеристиките на най-успешните пакети, техните силни страни, които трябва да присъстват в настоящата среда, както и техните недостатъци, за които трябва да се внимава да не се допуснат повторно. Анализирани са и начините за интегриране на компонентите на текущия софтуерен пакет в новата среда за моделиране. Резултатите са публикувани в [9].

Трета глава представя две възможности за оптимизация на ОМ модели. В раздели 3.2 и 3.3 са предложени нови, оптимизирани алгоритми за функциониране на ОМ и някои техни разширения. В раздел 3.4 са разгледани два глобални оператора за трансформиране на ОМ. Предложена е формална дефиниция на единия оператор и са разгледани условията, при които той не може да бъде приложен. Дефинирани са разширения на двата оператора, които налагат по-малко ограничения в сравнение със съществуващите. Резултатите са публикувани в [1, 2, 4, 5].

В глава 4 е разгледан въпросът за представимостта на мрежите на Кан с ОМ. Представена е универсална ОМ, която моделира функционирането и резултатите от работата на произволна мрежа на Кан. Резултатите са публикувани в [3].

В глава 5 са представени функционални и нефункционални изисквания към среда за моделиране с ОМ. Предложена е модулна архитектура на софтуерната ѝ реализация. Направен е анализ на потребителския интерфейс с цел оптимизиране на взаимодействието на потребителите със средата. Резултатите са публикувани в [7, 8, 10].

Глава 6 описва прилагането на разработената среда чрез създаване на ОМ модел, сравняващ различни модели на полупериодичен култивационен процес на *E. coli*. Моделът е симулиран с помощта на средата, описана в глава 5. Показани са възможностите за интегриране със средата за числово пресмятане MATLAB. Резултатите са публикувани в [6].

В заключението са обобщени резултатите от дисертационния труд. Предложени са насоки за бъдеща работа по разширения на теорията на ОМ и по развитието на софтуерната ѝ реализация.

Приложение А допълва глава 5 с описание на конкретните технологии и методи, избрани за реализацията на интегрираната среда. Включен е и детайлен обектно-ориентиран дизайн на средата. Разработваното приложение е описано подробно.

Някои използвани съкращения

ОМ – Обобщена мрежа;

МП – Мрежа на Петри;

ИМ – Индексирана матрица;

GN – Generalized Net – обобщена мрежа;

AT – Abstract transition – абстрактен преход;

GNTCFL – GNTicker Characteristic Function Language – език за характеристични функции и предикати, използван от GNTicker;

MVC – Model-View-Controller;

UML – Unified Modeling Language.

Кратко съдържание на дисертационния труд

Глава 1. Проблемна област

Обобщените мрежи са дефинирани през 1982 година. Те представляват разширение на мрежите на Петри, което обобщава различните други разширения, известни дотогава.

1.1 Мрежи на Петри

МП представлява двуделен ориентиран граф, в който има два вида възли - съответно за *преходи* (transitions) и за *позиции* (places). Преходът представя дискретно събитие, а позицията - условие, като в зависимост от посоката на дъгата то бива предусловие или постусловие на събитието. Позициите биват входни, изходни и вътрешни. МП съдържа динамични елементи, наречени *ядра* (tokens), които преминават по дъгите на графа и през преходите се придвижват от една позиция в друга. Така ядрата показват докъде е достигнал всеки от описаните чрез мрежата паралелно протичащи процеси. Поради нуждата от по-силни моделиращи възможности МП стават обект на повече от 20 различни разширения.

1.2 Обобщени мрежи

ОМ се дефинират по начин, по-различен от начините за дефиниране на други МП. За разлика от МП, в ОМ всяка позиция може да има не повече от една влизаща дъга и не повече от една излизаща. Всеки преход трябва да има поне една входна позиция и поне една изходна, която може и да е същата.

Ядрата на ОМ се придвижват през преходите от входни позиции в изходни. Информацията на дадено ядро се съхранява в множество от *характеристики*. Ядрата влизат в мрежата с *начални характеристики*, а по време на движението си получават и нови характеристики, евентуално запазвайки предишните. Така те натрупват собствена *история*. Всяка позиция може да притежава *характеристична функция*, която задава нови стойности на характеристиките на ядрата,

които влизат в нея.

Към всеки преход може да бъде асоциирана матрица от предикати, която позволява детайлно описание на условията за движение на ядрата в различните възможни посоки на движение.

Освен дискретното време, в което протичат процесите в МП, в ОМ е добавена и абсолютна времева скала, по която може да се отчита времето на функционирането на мрежата.

С обобщена мрежа може да се опишат както разширенията на МП, така и на машината на Тюринг. Всички известни до момента разширения на ОМ са *консервативни*.

В първа глава на дисертационния труд е дадена формална дефиниция на ОМ, както и досегашният алгоритъм за движение на ядрата в ОМ. Описана е методология за конструиране на ОМ, състояща се от следните етапи:

1. определяне на статичната структура на моделирания процес;
2. отразяване на динамиката на моделирания процес;
3. функциониране на моделирания процес във времето.

Глава 2. Анализ на наличните средства за работа с ОМ

През годините се е работело по създаването на множество симулатори и визуални редактори, дефинирани са били формати за съхранение на ОМ модели и др. В глава 2 на дисертационния труд се анализират успешните съвременни софтуерни пакети за моделиране с ОМ. Направеният критичен анализ има няколко основни цели:

- разглеждане на функционалността на съществуващите средства за работа с ОМ и определянето на техните силни страни, които е необходимо да бъдат налични в една интегрирана среда за моделиране с ОМ;
- посочване на пропуски и недостатъци в реализациите, които трябва да бъдат преодоляни с разработването на нова среда;
- анализиране на възможностите за интегриране на съществуващи инструменти към единна среда за моделиране с ОМ.

Установено е, че при всеки от пакетите са били използвани съвременни към момента на създаването им софтуерни технологии, но не във всички случаи те са отразявали актуалното състояние на теорията на ОМ. Един от резултатите от обзора е, че се проследява по какъв начин развитието на теорията намира отражение в разработвания през годините софтуер.

В различните пакети се срещат редица добри идеи, например удобен потребителски интерфейс за визуално дефиниране на модели, интерактивна симулация, различни начини за визуализиране на резултатите от симулацията и много други.

Характерно за всички софтуерни пакети е, че реализират част от теорията на ОМ – основно елементи от дефиницията и алгоритъма за функциониране, но само малка или никаква част от другите аспекти.

Основен проблем на съществуващите софтуерни реализации е, че не са завършени. При всеки следващ опит е започвано почти отначало, с нов екип разработчици. Налице са различни програмни грешки. В почти всички пакети, с изключение на GN Lite, възможностите за разширяемост са ограничени или липсват. Макар и всеки от инструментите да притежава своите достойнства, те не предоставят цялостно решение, което може да адресира всички етапи от процеса на моделиране – създаване, редактиране, симулиране, наблюдение, оптимизация.

Изчерпателен исторически преглед на всички опити и постигнати резултати за програмна реализация на ОМ, не само на разгледаните, е представен в [9].

Глава 3. Оптимизация на алгоритми за изпълнение и оператори за модификация на модели

Първоначалната цел на дисертацията е създаването на интегрирана среда за моделиране с ОМ. В процеса на обмисляне на разработката беше достигнато до извода, че е необходимо да се извършат подобрения в две направления:

- по отношение на алгоритмите за функциониране на ОМ;
- по отношение на предложения от специалиста модел, представен

чрез ОМ.

Първата задача се разрешава чрез оптимизиране на досегашните алгоритми за движение на ядрата в преход и в ОМ. За целите на втората задача, чрез използването на глобални оператори над ОМ, потребителският вход може да бъде подложен на неколкократни трансформации и оптимизации.

3.1 Оптимизирани алгоритми за функциониране на ОМ

Една от важните задачи на функционирането на един симулатор на ОМ са двата алгоритъма за движение на ядрата, съответно в преход и в ОМ. Те определят семантиката на ОМ, както и ефективността на разработения софтуер. Алгоритъмът за движение на ядрата в ОМ е значително по-сложен от съответния алгоритъм за МП. В настоящия раздел едновременно са оптимизирани и двата алгоритъма за движение на ядра в ОМ – алгоритъм А за функциониране на преход в ОМ и алгоритъм В – за функциониране на ОМ.

Следва описание на оптимизираните алгоритми.

Алгоритъм А – общ алгоритъм за функциониране на преход в даден момент от време t_1

(А01) За всяка входна позиция се формират два списъка:

• списък с всички ядра, сортирани по приоритет в намаляващ ред;

• празен списък.

(А02) Конструира се празна ИМ R с размерност, равна на размерността на r (индексираната матрица с предикати на прехода). Присвоява се стойност 0 (съответстваща на стойност „$false$“) на всички елементи на R, които се намират на:

• ред, съответстващ на празна входна позиция (т.е. няма постъпило ядро),

• стълб, съответстващ на пълна изходна позиция (т.е. капацитетът й е 0),

• клетка, съответстваща на дъга с капацитет 0.

(А03) Позициите, които предварително са сортирани, се обхождат последователно по ред на намаляване на приоритета им. Започва се с позицията с най-висок приоритет, в която има поне едно ядро и

през която не е имало трансфер в текущата стъпка от време. За най-приоритетното ядро (от първия списък) се определя дали може да разцепи или не. Това зависи от динамичен оператор, дефиниран върху дадената ОМ. Ако такъв оператор не е дефиниран, се приема, че ядрото може да се разцепва толкова пъти, колкото е необходимо. Проверяват се предикатите, съответстващи на позициите в реда на ИМ R. Ако ядрото не може да се разцепва, проверките спират при намирането на първия предикат, чиято истинна стойност е различна от 0. В противен случай се изчисляват всички предикати в реда, за които съответните елементи на R не са нули.

(A04) Ядрото от стъпка **(A03)** се придвижва до всички изходни позиции, за които съответната клетка в R има стойност 1 (ако разцепването не е разрешено, позицията ще е най-много една). Ако ядро не може да премине през прехода в текущия интервал от време, то се премества във втория списък от ядра на съответната входна позиция. Ядрата, които са постъпили в позицията след момента на активиране на прехода, също се преместват във втория списък.

(A05) Изчисляват се стойностите на характеристичните функции за изходните позиции (една или повече), в които са постъпили ядра (според стъпка **(A04)**).

(A06) Поставят се стойности „0“ във всички редове на R, за които съответната входна позиция (от която пристига ядро от стъпка **(A04)**) вече е празна; във всички колони на R, които съответстват на изходни позиции, които са запълнени в резултат на влизане на ядро на стъпка **(A04)**; във всички клетки, съответстващи на дъги между дискутираната входна позиция и онези изходни позиции, за които капацитетът на дъгата е станал 0 в резултат на движението на ядро.

(A07) За всяка входна позиция текущият брой ядра в нея се намалява с 1 за всяко ядро, което трябва да я напусне в текущата стъпка. Ако текущият брой ядра в дадена входна позиция стане нула, на елементите на съответния ред в индексираната матрица R се присвоява стойност „0“.

(A08) Капацитетите на всички позиции, в които е влязло ядро, определено в стъпка **(A03)**, се намаляват с 1. Ако бъде достигнат максималният брой ядра за дадена изходна позиция, елементите на съот-

ветната колона в R се променят на „0“.

(A09) Капацитетите на всички дъги, по които е преминало ядро, се намаляват с 1. Ако капацитетът на дадена дъга достигне 0, съответният елемент в R получава стойност 0.

(A10) Ако все още има ядра във входните позиции, които трябва да бъдат преместени, има незапълнени изходни позиции и има дъги с ненулеви капацитети, то алгоритъмът продължава със стъпка **(A11)**; в противен случай се отива на стъпка **(A13)**.

(A11) Текущата стойност t' на моделното време се увеличава с t^o ($t' = t_1 + t^o$).

(A12) Проверява се достигнато ли е време $t_1 + t_2$. Ако отговорът на въпроса е „не“, се отива към стъпка **(A03)**, в противен случай се отива в **(A13)**.

(A13) Край на функционирането на прехода.

Алгоритъм B – общ алгоритъм за функциониране на ОМ

(B01) На текущото време t_T се присвоява стойност T.

(B02) Всички неизползвани до момента ядра α, за които $\theta_k(\alpha) \leq t_T$, се поставят в съответните им входни позиции на мрежата.

(B03) Конструира се абстрактният преход на мрежата. Той е обединение на всички активни преходи в даден момент от време.

(B04) Избират се всички преходи, за които първият времеви компонент е равен на текущия момент от време t_T.

(B05) Изчисляват се типовете ($\square$) на всички преходи, намерени на стъпка **(B04)**. Използва се следният метод:

- идентификаторите на всички позиции, участващи в типа на прехода, се заместват с 0, ако в съответната позиция няма ядра в текущия момент, или с 1, в противен случай;
- изчислява се стойността на така получения булев израз.

(B06) Към абстрактния преход се добавят всички преходи от **(B05)**, за които $\square = 1$. Извършват се следните стъпки:

- Ако не е възможно дадена позиция да промени приоритета си по време на симулация, за всеки преход, който трябва да бъде добавен, се извършва следното:

  - Сортират се съответно входните и изходните позиции на прехода, ако никога не са били сортирани от началото на симулацията до момента.
  - Сливат се сортираните списъци с позиции, започвайки от най-краткия.

- Ако е възможно позиция да промени приоритета си по време на симулация:
  - Сортират се всички непразни входни позиции и всички непълни изходни позиции в АТ.
  - Сортират се всички непразни входни позиции и всички непълни изходни позиции на всеки преход, който трябва да бъде добавен към абстрактния.
  - Сливат се сортираните списъци с позиции, започвайки от най-краткия. Несортираните секции се закачат след сортираните.

(B07) Ядрата в изходните позиции, които не са входни за друг преход в мрежата, напускат ОМ.

(B08) Прилага се една стъпка от алгоритъм А върху абстрактния преход.

(B09) От абстрактния преход се премахват всички преходи, които не са активни в текущия момент от време. Подредбата на позициите в АТ, както и подредбата на позициите от премахнатите преходи, се запазват.

(B10) Текущото време се увеличава с t^o.

(B11) Проверява се дали текущият момент от време е по-голям от $T + t^*$.

(B12) Ако отговорът на въпроса от **(B11)** е отрицателен, се отива на стъпка **(B02)**, в противен случай се преустановява функционирането на обобщената мрежа.

Предложените оптимизации запазват оригиналната структура и четливост. В повечето случаи се проверяват по-малко предикати, сортират се по-малко позиции, оптимизирани са матриците на абстрактните позиции. Функционирането и резултатите от работата на дадена ОМ са

едни и същи, независимо дали е използван старият или новият алгоритъм.

3.2 Оптимизация на алгоритмите за движение на ядрата в различните видове интуиционистки размити ОМ

През годините са дефинирани множество консервативни разширения на обобщените мрежи, например цветни ОМ, ОМ с глобална памет, ОМ с оптимизационни компоненти и много други. Едни от най-често използваните и важни за практиката разширения на ОМ са интуиционистки размитите ОМ (ИРОМ). По тази причина ще бъдат приложени оптимизации на алгоритмите за движение на ядрата и в тях.

При интуиционистки размитите ОМ стойностите на функцията f, която изчислява логическите стойности на предикатите, принадлежат на множеството $\{\langle a, b\rangle | a, b, a + b \in [0, 1]\}$, вместо на $\{false, true\}$ (или $\{0, 1\}$). Нека над съвкупността от съждения S е дефинирана оценъчната функция $V : S \to [0, 1] \times [0, 1]$:

$$V(p) = \langle \mu(p), \nu(p)\rangle$$

V задава степените на вярност и невярност на съжденията от S.

Предложените оптимизации са приложени върху алгоритмите за движение на ядрата в ИРОМ от първи, втори, трети и четвърти тип.

3.3 Формални дефиниции, модификации и свойства на два оператора над ОМ

Всеки оператор съпоставя на дадена ОМ нова ОМ със специфични свойства. Глобалните оператори трансформират съгласно определена процедура цяла ОМ или всички нейни компоненти от даден тип. От практическа гледна точка са интересни операторите G_2 и G_6.

Оператор G_2 променя формата и структурата на дадена ОМ E като я свива до единичен преход. От всички позиции се запазват само тези, които са входни или изходни за мрежата E.

Оператор G_6 променя формата и структурата на дадена ОМ E, свивайки всяка нейна „линейна“ подмрежа в единичен преход. Ще наричаме линейна ОМ всяка ОМ, за която изходите на първия преход

съвпадат с входовете на втория, изходите на втория с входовете на третия и т.н. G_6 опростява статичната структура на даден ОМ модел.

В литературата се срещат само абстрактни и неформални дефиниции на разгледаните два оператора. Слабости в съществуващите дефиниции, макар и свързани с крайни случаи, могат да дадат отражения върху реализирането на практически модели. Необходимо е да бъдат формално дефинирани модификации на сегашните оператори, които да не променят резултатите от работата на мрежата в нито една стъпка от функционирането й. Изискване към новите оператори е да покриват всички специални случаи, които не са адресирани от оригиналните дефиниции. По този начин специалистът ще може автоматизирано да извършва опростявания на ОМ модели.

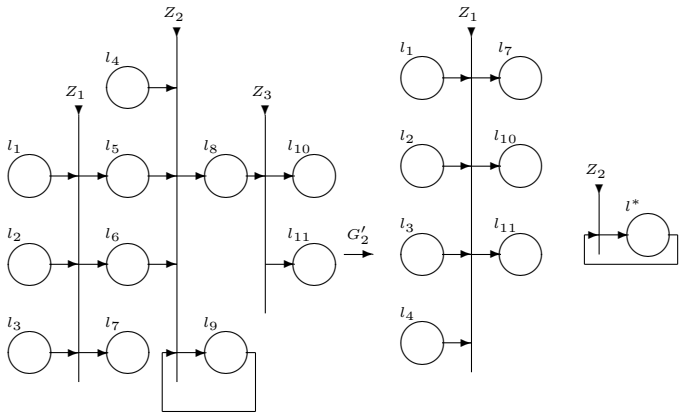
Дефинирането на алгоритъм за прилагане на G_2 (и следователно програмната реализация на G_2) не е тривиална задача. Има много случаи, които трябва да бъдат взети предвид. Например даден предикат може да проверява дали дадена позиция съдържа ядро, но тази позиция да е премахната от G_2. По тази причина в настоящия раздел се въвежда нов глобален оператор G_2'. Изискването е той да запазва оригиналната идея на G_2, но да може да бъде дефиниран с по-прост алгоритъм.

Нов глобален оператор G_2'

Оператор G_2' свива цяла ОМ E до мрежа E' с два прехода (фиг. 1). Първият има същата топологична структура като прехода, получен при прилагането на G_2. Вторият преход има една позиция l^*, която е едновременно входна и изходна. Първоначалната ОМ се съхранява като характеристика в ядро γ, което се движи от l^* в l^*. На всяка стъпка от функционирането на E една стъпка от алгоритъма за движение на ядра в ОМ се прилага върху E от характеристичната функция на l^*. Ядрата в новата мрежа E' биват синхронизирани с текущото състояние на ядрата в E. За да е възможно ядрата да бъдат преместени в изходните позиции на E' още в същата стъпка, в която Φ_{l^*} е изпълнена, новата позиция l^*, преходът Z_2, към който принадлежи, както и ядрото γ имат най-високия приоритет сред всички компоненти на E'. Разцепването и сливането на ядра не трябва да бъде разрешено в E, т.е. оператори $D(2,1)$ и $D(2,4)$ не трябва да бъдат дефинирани за

мрежата.

Фигура 1 илюстрира трансформирането на примерна ОМ от оператор G_2'. В раздел 3.4 на дисертационния труд е дадена формална дефиниция на G_2'.



Фигура 1: Примерна ОМ и резултатът от прилагането на G_2' върху нея

Формализация на G_6

С pr_iX е прието да се означава i-та проекция на n-мерното множество X, $n \in \mathbb{N}$, $n \geq 1$, $1 \leq i \leq n$. По-общо: $pr_{i_1,i_2,\ldots,i_s}X = pr_{i_1}X \times pr_{i_2}X \times \cdots \times pr_{i_s}X$, където $1 \leq i_1 \leq i_2 \leq \cdots \leq i_s \leq n$, n е размерността на X и s е естествено число.

Следва съкратена дефиниция на G_6. В раздел 3.4 е дадена пълна дефиниция. G_6 преобразува всяка линейна последователност от преходи $Z_0, ..., Z_k$ в E до преход, имащ следния вид:

$$\langle L', L'', t_1, t_2, r, M, \square \rangle,$$

където:

$$L' = pr_1 Z_0$$

$$L'' = pr_2 Z_k$$

$$t_1 = pr_3 Z_0$$

$$t_2 = \sum_{i=0}^{k} pr_4 Z_i$$

$$\square = pr_7 Z_0$$

Предикатната матрица и матрицата на капацитетите са композиции на съответните матрици на премахнатите преходи. Всеки елемент на r е дизюнкция от конюнкциите на предикатите по всеки възможен път между съответните две позиции, докато при M той е максимумът на минималните капацитети на всеки възможен път.

Характеристичната функция Φ за всяка изходна позиция $l_j \in pr_2 Z$ изпълнява характеристичните функции на всички позиции по всички възможни пътища между позиция l_i, откъдето е дошло ядрото, за което се изпълнява Φ, и текущата изходна позиция l_j, за които пътища конюнкцията от предикатите е истина.

Сравнение на функционирането на E и $G_6(E)$

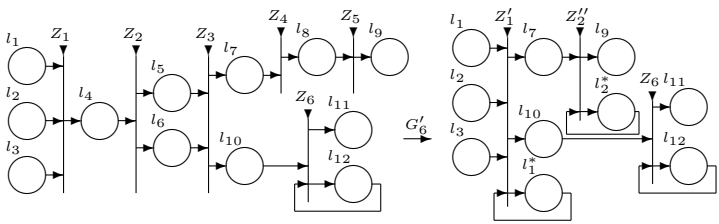
Прилагането на така дефинирания оператор G_6 върху някои ОМ може да доведе до получаването на неконсистентни модели. Ако за дадена ОМ някой от нейните предикати или характеристични функции зависи от позиция, премахната от оператора при неговото прилагане, този предикат или функция ще бъде недефиниран. Ако някое от следните условия не е изпълнено, резултатите от работата на E и $G_6(E)$ може да не съвпадат:

- Предикатите и характеристичните функции в линейна подмрежа не трябва да зависят от стойности на характеристики, получени от характеристична функция на премахната позиция.
- Ако характеристична функция Φ_l е изпълнена за ядро α, Φ_l не трябва да променя характеристиките на ядра, различни от α.
- Типовете на преходите Z_1, ..., Z_k трябва да бъдат дизюнкции, т.е. $\vee(l_1, ..., l_n)$, където $l_1, ..., l_n$ са входните позиции на съответния преход.
- Приоритетите на всички вътрешни (подлежащи на премахване) изходни позиции на даден преход трябва да бъдат равни.
- Капацитетите на всички вътрешни изходни позиции (съответно премахнати влизащи дъги, а също и излизащи дъги) на даден

преход трябва да бъдат равни.

Нов глобален оператор G'_6

Оператор G'_6 свива всички линейни участъци в дадена ОМ E по същия начин като G_6, но с една съществена разлика. Към всеки преход, който заменя линейна последователност от преходи, се добавя цикъл, който събира влизащите ядра. Ако линейната подмрежа има $k+1$ прехода, ядрата преминават $k-1$ пъти през новата позиция, преди да напуснат цикъла и да влязат в някоя изходна позиция. По този начин функционирането на E във времето не се променя. Характеристичната функция на новата позиция комбинира всички характеристични функции, предикати и типове на преходи на всички премахнати компоненти. За всяко ядро тя имитира процеса по преминаване през различните позиции по пътя. За да постигне това, тя добавя характеристика, наречена *virtual_host*, към всяко ядро, което постъпва в позицията. Стойността на новата характеристика е несъществуващата позиция, в която ядрото би влязло, ако операторът не е бил приложен. Новият цикъл също така опростява трансформирането на предикатите и характеристичните функции. Ако даден предикат или функция зависи от премахната позиция, той ще бъде модифициран така, че да реферира позицията от цикъла и да филтрира ядрата в нея по *virtual_host*, както е описано по-надолу в раздела.



Фигура 2: Примерна ОМ (вляво) и резултатът от прилагането на G'_6 върху нея (вдясно)

Фиг. 2 показва примерна ОМ E с две линейни последователности от преходи преди и след прилагането на $G'_6(E)$. В раздел 3.4 на дисерта-

ционния труд е дадена подробна формална дефиниция на предложения оператор.

Заключение

В глава 3 на дисертацията са оптимизирани и двата алгоритъма за движение на ядра в ОМ – алгоритъм А за функциониране на преход и алгоритъм В – за функциониране на ОМ. Предложените оптимизации запазват оригиналната структура и четливост. Новите алгоритми не променят функционирането и резултатите от работата на дадена ОМ.

Разгледани са два глобални оператора над ОМ, които променят структурата на даден модел. Дефиниран е нов глобален оператор G_2'. Предложена е формална дефиниция на съществуващия оператор G_6, разгледани са условията, при които не може да бъде приложен операторът G_6 и са сравнени функционирането и резултатите от работата на дадена мрежа E и $G_6(E)$. Дефиниран е нов глобален оператор G_6', който не налага ограниченията на G_6. Двата нови оператора са разширение на съответните съществуващи, които чрез еквивалентни преобразувания запазват функционирането на мрежата във времето и резултатите от нейната работа за значително по-широк клас ОМ спрямо предишните оператори.

Глава 4. Представимост на мрежи на Кан чрез обобщена мрежа

Въведение

Едно от най-значимите теоретични предимства на ОМ е способността им да моделират голямо разнообразие от други моделни средства. Така симулаторът може да се използва като мощно средство, интегриращо моделирането не само с ОМ, но и с много други популярни средства.

Дефинирането на универсални ОМ, представящи работата на други моделиращи средства, е удобен начин разнородни модели да бъдат симулирани със софтуера за ОМ. По този начин специалистът може да се възползва от преимуществата на ОМ пред другите моделиращи средст-

ва. Вместо да се налага разработката и използването на специализиран софтуер за симулация, специалистът може да работи в рамките на ОМ и съответната теоретична или приложна област.

Едно ключово възможно приложение на разработената в рамките на този дисертационен труд среда за моделиране е симулирането на процеси, комуникиращи през опашкови канали, например в областта на обработката на цифрови сигнали. В настоящата глава се разглежда един мощен механизъм за описание на такива процеси и се обяснява как този механизъм е представим чрез ОМ, което прави възможно симулирането на такива модели чрез софтуера за симулация на ОМ.

Мрежите на Кан (Kahn Process Networks) са изчислителен модел, който описва комуникацията между процеси през неограничени канали от тип „първи влязъл, първи излязъл“ (FIFO). Операциите, свързани с четене от канал, са блокиращи, докато тези, свързани с писане, не са. Даден процес не може да провери дали даден канал е празен. Каналите биват входни, изходни и вътрешни. Всеки процес p в една мрежа на Кан е детерминистичен, т.е. по една и съща последователност от въвеждане на входни данни p генерира едни и същи изходни данни. Нито времето, нито редът на изпълнение не влияят на резултата.

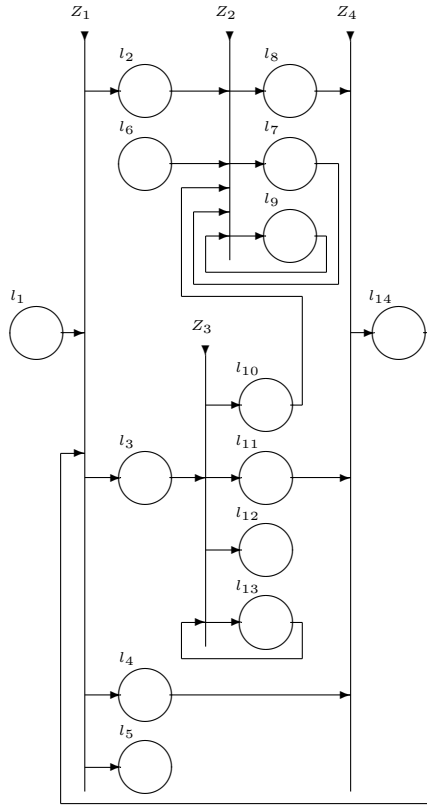
ОМ, описваща произволна мрежа на Кан

В глава 4 на дисертационния труд се дава формална дефиниция на мрежа на Кан. Показва се, че ОМ могат адекватно да описват мрежи на Кан. Дадена е формална дефиниция на ОМ, представяща функционирането и резултатите от работата на произволна мрежа на Кан.

Фиг. 3 показва графичната структура на конструираната ОМ.

В позиция l_1 постъпва ядро π, съдържащо като характеристика конфигурацията на всички процеси. В зависимост от текущото действие на всеки процес, π се разцепва на четири ядра π_i, $1 \leq i \leq 4$, които представят четири категории от процеси: четящи, пишещи, изчисляващи и прекратени процеси. Разцепването и сливането на ядра трябва да бъде разрешено за мрежата, т.е. оператори $D(2,1)$ и $D(2,4)$ трябва да бъдат дефинирани за E.

Ядрото π_1, съдържащо процесите, чието текущо действие е четене,



Фигура 3: Графична структура на универсална ОМ, представяща мрежа на Кан

преминава от l_2 в l_9, където всеки процес чака да пристигнат данни от съответния входен канал. В даден момент ядро ι, съдържащо данни за четене, постъпва в позиция l_6. ι отива в l_7, където се пазят всички непрочетени елементи от данни от входните канали. Тъй като са налични както четящ процес, така и данни от съответния входен канал, π_1 и ι се придвижват до l_8, където те се сливат в π_1'. Ако съществуват други блокирани четящи процеси или има входни данни, за които няма процес, който да ги прочете, те продължават да чакат съответно в l_9 и l_7.

В случай на четене от вътрешен канал c, а не от входен, в даден момент ядро δ постъпва в позиция l_{10} – или непосредствено след като π_1 влиза в l_9, или, ако c е празен, след като някой друг процес пише в него. Характеристика на δ са новите данни на вътрешните канали. Ядрата π_1 и δ отиват в l_8, където те се сливат до π_1' и конфигурациите на процесите (в π_1') и на вътрешните канали (в γ) се променят.

Ядро π_2 от позиция l_3 съдържа процесите, чието текущо действие е писане. Този път няма изчакване и ядрото се придвижва директно в l_{11}, където конфигурацията на процесите се обновява. Данните, записани от p, се съдържат в ядро ρ, което напуска E през изходната позиция l_{12}. Ако процес пише във вътрешен канал c, а не в изходен, изходните данни се добавят към конфигурацията на вътрешните канали, съхранявана в ядро γ в позиция l_{13}. Ако съществува процес, който чака да прочете елемент от c, на следващата стъпка ядро δ, съдържащо новите данни от c, отива в позиция l_{10}.

За представянето на процесни мрежи с ограничения на каналите

Дефиницията на мрежа на Кан позволява неограничени канали, но приложенията от реалния свят налагат ограничения в броя на неконсумираните елементи от даден канал. Конструираната ОМ може лесно да бъде променена да поддържа процесни мрежи с ограничения на каналите.

Стратегия за планиране (scheduling strategy) може лесно да бъде интегрирана в E'. Предикатите на преходите могат така да бъдат променени, че динамично да регулират капацитетите на каналите и да определят кои процеси могат да бъдат изпълнени на дадена стъпка. Конструираната ОМ може да бъде използвана за тестване на съществуващи и нови стратегии.

Предимства на ОМ пред мрежите на Кан

Така конструираната ОМ е универсална за всички мрежи на Кан с неограничени канали. Обобщената мрежа е редуцирана, т.е. дори прост клас ОМ е достатъчен, за да опише мрежите на Кан.

Използването на ОМ, за да се моделира реален процес, за който обикновено се използва мрежа на Кан, дава множество предимства, понеже ОМ са по-детайлно моделиращо средство. Ядрата в ОМ имат история на характеристиките, така че всички промени в конфигурациите на процесите и каналите могат да бъдат проследявани. Ако реален процес е представен чрез ОМ, вместо с мрежа на Кан, моделиерът може да използва глобалния времеви компонент на ОМ, така че времето на процеса може да бъде съотнесено към времевата скала. За разлика от мрежите на Кан, ОМ поддържат различни видове време. ОМ моделът също може лесно да бъде интегриран с нова ОМ, която управлява реда на изпълнение на процесите и капацитетите на каналите в процесни мрежи с ограничени канали.

Разработената ОМ може да се използва за софтуерна симулация на мрежи на Кан с помощта на интегрираната среда за моделиране с ОМ.

Глава 5. Софтуерен пакет за работа с ОМ

Една от основните цели на настоящата дисертация е проектирането и реализирането на удобна интегрирана среда за визуално конструиране на ОМ модели и тяхната симулация. Разработката на цялостна интегрирана среда за моделиране с ОМ е мащабна задача, която включва редица аспекти от софтуерното инженерство, като внимателен и детайлен анализ на изискванията към такава среда, изработването на гъвкава архитектура, състояща се от множество модули, и не на последно място подбор на подходяши средства за програмната й реализация.

Изисквания към софтуерното приложение

За да се постигне висока полезност на софтуера, са дискутирани и обобщени потребителски изисквания от различни заинтересовани лица (stakeholders).

С помощта на интегрирана среда за моделиране с ОМ симулацията на модел може да се извърши на следните стъпки:

1. Чертае се изготвената статична структура на модела чрез удобен за потребителя графичен редактор. Въвеждат се параметрите на

модела: условия за преминаване на ядрата през преходите, начални стойности на характеристиките на ядрата, характеристични функции на ядрата, капацитети и приоритети на позициите и дъгите, условия за сливане и разцепване на ядра, параметри за време и продължителност и други.

2. Симулацията се стартира с възможност за управление от потребителя: начало на симулация, постъпково и автоматично изпълнение, пауза, продължаване на симулацията.

3. По време на симулацията потребителят може да следи моментното състояние на модела, както и историята на характеристиките. Възможно е връщане на модела в предишно състояние по време на пауза. Всички резултати от симулацията могат да се запишат с цел възпроизвеждане на същата симулация и по-детайлно изследване на резултатите от нея.

4. Моделите могат да се запазват в XGN формат, а също и да се експортират до TeX и графични формати като SVG.

Изискванията за подобрение на основния компонент на GN Lite – GNTicker – са следните:

- Интегриране с MATLAB. GNTicker предлага възможност потребителят да дефинира произволни предикати и характеристични функции в OM, но към момента не са публикувани библиотеки за GNTicker, предлагащи наготово сложни изчисления, например числени методи за решаване на диференциални уравнения.
- Реализация на новите алгоритми за функциониране на преход и OM.
- Поддръжка на JavaScript. Поддържаният до момента програмен език за предикати и характеристични функции GNTCFL не е особено подходящ за софтуерен продукт, целящ широка популярност. JavaScript е масово използван и лесен за изучаване език.

Дефинирани са и следните нефункционални изисквания: използваемост, лесно научаване, лесно внедряване и портативност, платформена

независимост, разширяемост, възможност да се включат нови разработчици, бързо пускане на работеща версия и задоволителна производителност.

Избор на методи и технологии

Настоящата интегрирана среда за моделиране с ОМ, както и новата версия на GNTicker и TickerServer, са написани на Java. За работна среда (IDE) е използван Eclipse. За изграждането на потребителския интерфейс на визуалната среда е използван Standard Widget Toolkit (SWT). Build процесът е автоматизиран с Apache Ant. За реализирането на поддръжката на JavaScript като език за предикати и характеристични функции в ОМ модели е използван Apache Rhino.

При проектирането на потребителския интерфейс на визуалната среда за ОМ са разгледани основни шаблонни поведения на потребителите, дефинирани от J. Tidwell.

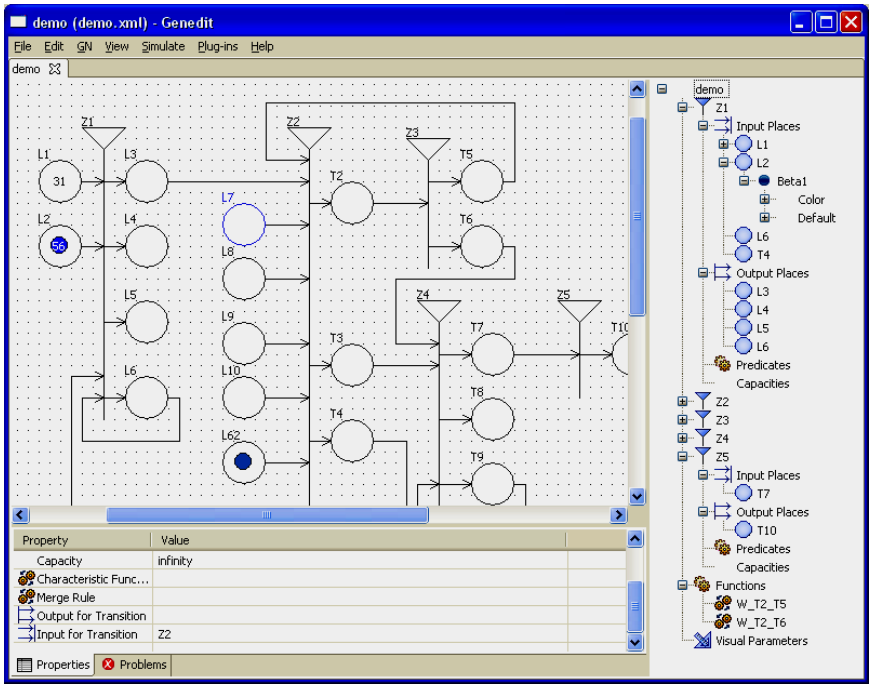
На фиг. 4 е показан главният прозорец на интегрираната среда. Той се състои от няколко основни изгледи:

Основният изглед на ОМ моделите е **графичният изглед**. В него се изобразява статичната структура на ОМ, както и ядрата и някои техни характеристики. Графичният изглед позволява избирането на изобразени обекти, като селекцията е видима и от другите изгледи.

Йерархичен изглед – представя ОМ модела чрез дървовидна структура. Коренът представя самата ОМ. Следващите нива на йерархия представят съответно преходите, позициите (входни и изходни за всеки преход), ядрата в тези позиции, характеристиките на ядрата, историята на характеристиките. Подобен изглед се реализира за първи път в настоящата интегрирана среда.

Изглед на свойствата – в таблица с две колони се визуализират свойствата на текущо избрания обект в графичния или дървовидния изглед. Показаните стойности могат да бъдат редактирани от потребителя.

Друг важен изглед е **изгледът на функциите** – използва се за дефиниране на предикати в матриците на преходите и характеристични функции за позициите.



Фигура 4: Главен прозрец на GN IDE

Изчертаването на ОМ модел в настоящото приложение може да се извършва по няколко основни начина, като най-удобният е чрез разпознаване на жестове. Потребителят влачи курсора на мишката по същия начин, по който би рисувал с молив върху лист хартия. След това приложението прилага алгоритъм за разпознаване на обекта, който трябва да бъде конструиран и избиране на оптималните размери и координати.

Архитектура на интегрираната среда за моделиране с ОМ

Основна цел, която се преследва при проектирането на интегрираната среда за моделиране, е обособяването на отделни компоненти със слаби ациклични зависимости между тях, с възможност за повторно използване от страна на бъдещи нови компоненти.

Първият слой е универсален за широк клас приложения и в него няма елементи, свързани с обобщените мрежи. Съдържа Model-View-Controller-библиотека и модул за работа с приставки (т.нар. plug-ins).

Модулът за поддръжка на разширения Plug-in Framework осигурява лесен механизъм за добавяне на нова функционалност към приложението. Тази възможност се използва и за основните модули на продукта, с цел да се елиминират цикличните зависимости между модулите.

В следващия слой е представянето на OM модели. Той използва базовата логика от подмодула Base Model на MVC-библиотеката и въвежда само функционалност, специфична за OM.

След като вече е уточнено представянето на OM, в следващия слой се намират различни приложни модули.

При проектирането на настоящото приложение са използвани множество шаблони за дизайн, дефинирани от Erich Gamma et al. Решенията са представени с UML диаграми.

Заключение

В глава 5 е описано разработването на интегрирана среда за моделиране с OM към пакет GN Lite, както и нова версия на симулатора за OM GNTicker. Новият софтуер реализира постигнатите в предходните глави теоретични резултати, благодарение на което има редица предимства пред останалите пакети. Използвани са новите, по-ефективни алгоритми за движение на ядрата в OM, описани в глава 3. Дефинираните от потребителите OM модели могат да бъдат трансформирани с помощта на разработените глобални оператори, с цел оптимизиране на графичната им структура и в същото време запазвайки резултатите от работата им. С разработената интегрирана среда за моделиране с OM потребителят може да симулира и други видове модели, в частност мрежи на Кан, без да се налага да изменя програмния код.

По време на работата по настоящия софтуерен пакет бяха срещнати различни трудности. Разработването на пълнофункционална интегрирана среда за моделиране е амбициозна и трудоемка задача. Представеният софтуер е разработван самостоятелно, а не от екип разработчици.

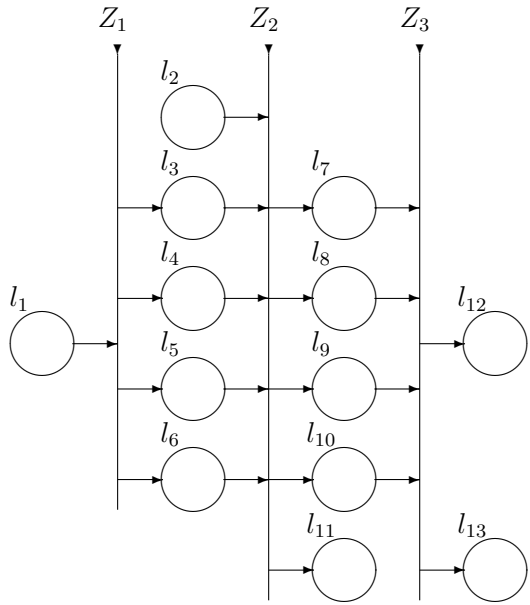
При получаването на нови изисквания по време на работа беше необходимо да се внимава да не се допусне т.нар. „scope creep“, т.е. неконтролируемо нарастване на обхвата на проекта. Описаният в настоящия дисертационен труд дизайн на интегрирана среда за моделиране с ОМ, а също и компонентно-базираната архитектура на GN Lite, позволяват лесно да се добавя нова функционалност.

Глава 6. Дефиниране на тестов модел и валидиране на разработената среда

Целта на глава 6 на дисертационния труд е демонстрирането на възможностите на разработената интегрирана среда за моделиране с ОМ чрез конструирането и симулирането на модел на реален процес. Избран е конкретен процес в областта на биотехнологиите, за който е необходимо извършването на сложни математически изчисления. За разработения ОМ модел от голяма полза е реализираната връзка със средата за числово пресмятане MATLAB, а също и възможностите за визуализация на данните от симулацията.

Апаратът на ОМ е използван, за да бъде създаден модел за сравнение на различни математически модели на полупериодичен култивационен процес на *E. coli*, който избира най-добрия модел на базата на определени критерии. Изграждането на ОМ модела е в сътрудничество със специалист по биотехнологии, като за база се използват разработени от нея математически модели на *E. coli* в среда MATLAB. Цел на разработената ОМ е интегрирането на няколко съществуващи разработки на MATLAB в един ОМ модел, който да послужи за тестване на разработената среда. Математическият модел, на който е базиран ОМ моделът, е представен със система от обикновени диференциални уравнения, описващи основните променливи на култивационния процес - биомаса и субстрат. За целите на симулацията са използвани три различни математически модела, идентифицирани с различни оптимизационни техники, както и четвърти, тестов модел.

На фиг. 5 е показана статичната структура на конструираната ОМ. Разцепването и сливането на ядра трябва да е позволено за мрежата, т.е. оператори $D(2,1)$ и $D(2,4)$ трябва да бъдат дефинирани за E.



Фигура 5: Статична структура на ОМ модел за сравнение на математически модели на полупериодичен култивационен процес на *E. coli*

Ядро α влиза в ОМ, съдържащо като характеристика четири математически модела, описани в раздел 6.1.2 на дисертационния труд.

В първия преход ядрото α се разцепва на четири ядра, съответстващи на всеки един от четирите модела. Всяко ядро е с различен цвят. Основните начални характеристики на полупериодичен процес на култивация на *E. coli* са дефинирани в ядро β в позиция l_2.

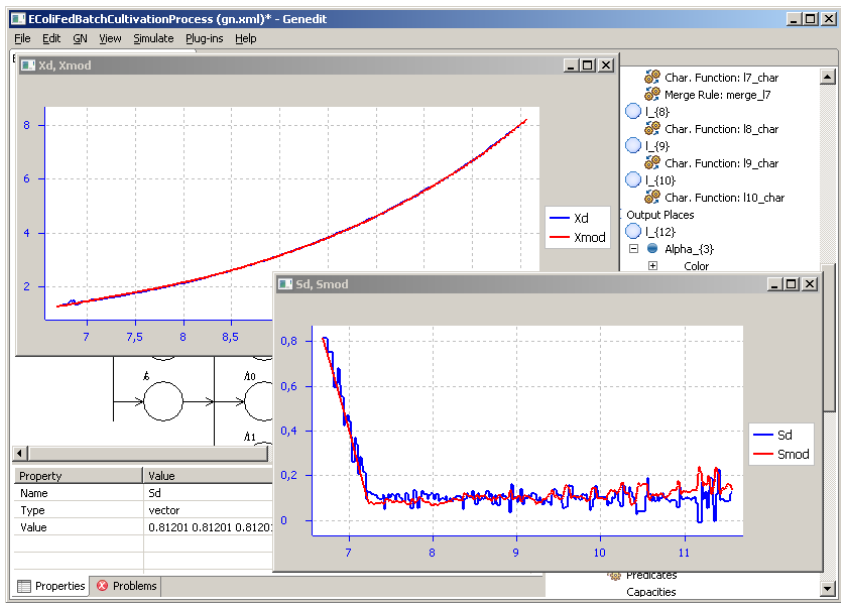
В следващия преход - Z_2 - всеки модел бива проверен за коректност според дефинираните интервали за всеки параметър. Моделите, които преминават успешно проверката, се придвижват до съответната изходна позиция (l_7, l_8, l_9, l_{10}). Останалите напускат мрежата през позиция l_{11}. В примерния случай три модела са преминали проверката, а четвъртият отива в l_{11}, за да напусне мрежата.

Ядрото β се разцепва за всяка изходна позиция, съответстваща на валиден модел. След прехода Z_2 двойките ядра в позиции l_7 до l_{10} се сливат.

За всяко ядро в l_i, $7 \leq i \leq 10$ характеристичната функция Φ извиква MATLAB функцията *simul*. В MATLAB среда дефинираната система от нелинейни диференциални уравнения се решава за всяко множество от параметри на модели. Функцията връща три стойности: J, X_{mod} и S_{mod} и ги добавя към характеристиката на съответното ядро.

В зависимост от стойността на J, в прехода Z_3 се избира ядрото с най-малка грешка и то се придвижва до позиция l_{12}. Останалите ядра се преместват в l_{13}. Ядрото в l_{12} е математическият модел на разглеждания процес с най-висока степен на точност.

ОМ моделът е симулиран компютърно с GN Lite. С разработената в тази глава ОМ са показани възможностите на интегрираната среда за моделиране с ОМ GN IDE, разработена към настоящия дисертационен труд, като особено внимание е обърнато на апробирането на връзката с MATLAB с голям обем реални данни. По време на симулацията ОМ моделът се обръща към средата MATLAB, за да се реши системата от диференциални уравнения. Резултатите се връщат обратно в GN Lite



Фигура 6: Резултати от симулацията

и на края на симулацията средата изчертава измененията на главните променливи на процеса на модела с най-висока степен на точност. В GN IDE потребителят може във всеки момент от симулацията да види текущата стойност на дадена характеристика (чрез посочване на съответното ядро с мишката или чрез избирането на характеристиката чрез дървовидния изглед). За по-голяма нагледност GN IDE дава възможност да бъдат изчертавани диаграми, показващи стойностите на избрани от потребителя характеристики. На фиг. 6 е показана динамиката на главните променливи на процеса (биомаса и субстрат). Реалните експериментални данни са сравнени с тези, предсказани от модела.

Заключение

Обобщените мрежи са мощен инструмент за моделиране и оптимизация на паралелни и конкурентни процеси, обобщаващ различните разширения на мрежите на Петри. Създадени са стотици ОМ модели на процеси от различни области. Разработването на цялостна интегрирана среда за моделиране и симулация с ОМ, каквато се явява разработваният в ИБФБМИ на БАН и ТУ-Дрезден пакет GN Lite, показва на практика предимствата на ОМ и осигурява използването на пълните им възможности.

С настоящия дисертационен труд е постигната целта да се проектира и реализира основен компонент на GN Lite, а именно интегрирана среда за моделиране с ОМ, както и нова версия на симулатора GNTicker. Това отваря вратите към широкото използване на новия пакет. Успоредно с това са постигнати нови резултати в теорията на ОМ. Разработени са и нови ОМ модели - както в теоретично направление (представимостта на мрежи на Кан с ОМ), така и с практическа насоченост (в областта на биотехнологичните процеси).

В глава 2 е направен критичен анализ на съществуващите средства за моделиране с ОМ. На негова база са дефинирани някои основни изисквания към новата интегрирана среда.

В глава 3 са подобрени алгоритмите за движение на ядрата в ОМ и

техни разширения - четирите типа интуиционистки размити ОМ. Дадени са формални дефиниции и са изследвани свойствата на няколко глобални оператора над ОМ. Дефинирани са нови глобални оператори, които опростяват структурата на дадена ОМ, запазвайки нейното функциониране и резултатите от работата ѝ.

В глава 4 е разработена обобщена мрежа, която представя функционирането и резултатите от работата на произволна мрежа на Кан. Това позволява чрез разработената интегрирана среда за моделиране с ОМ да се симулират и мрежи на Кан, което дава възможност на специалистите да се възползват от по-богатия набор от моделиращи възможности на ОМ.

Глава 5 разглежда проектирането и реализирането на интегрираната среда за моделиране с ОМ, една от целите на настоящия дисертационен труд. Описани са функционални и нефункционални изисквания към средата и към новата версия на симулатора GNTicker. Предложена е модулна архитектура на интегрираната среда.

Глава 6 демонстрира възможностите на разработената интегрирана среда за моделиране с ОМ чрез дефинирането на тестов ОМ модел, сравняващ различни математически модели на полупериодичен култивационен процес на бактерията *E. coli*. Акцентът пада върху интеграцията със средата за числово пресмятане MATLAB.

През цялото време на разработка интегрираната среда за моделиране е демонстрирана както пред български, така и пред чуждестранни учени, нейни потенциални бъдещи потребители. Приложението е анонсирано на Tenth International Workshop on Generalized Nets [7] и заедно с направения за първи път хронологичен обзор на всички публикувани софтуерни пакети за ОМ е докладвано пред Третата годишна сесия на секция „Информатика“ при Съюза на учените в България [8] [9]. На конференция на IEEE по интелигентни системи разработената универсална ОМ, представяща произволна мрежа на Кан, беше компютърно симулирана с GN Lite [3]. На юбилейния симпозиум по ОМ софтуерният пакет беше показан и пред учени от Австралия, Великобритания, Полша и други страни. Приложението беше представено в ИБФБМИ на БАН по време на докторантски курс по обобщени мрежи, както и на студенти от различни университети.

Проведените срещи спомогнаха както за популяризирането на пакета, така и за получаването на навременна обратна връзка, която е от особено значение за удовлетворяване на изискванията на потребителите.

На базата на настоящата работа могат да се определят следните насоки за бъдещо развитие:

- Формализация и програмна реализация на нови аспекти от теорията на ОМ.
- Добавяне на нова функционалност към софтуерния пакет. Поставени са амбициозни цели, затова е необходимо новите изисквания да бъдат приоритизирани внимателно, отчитайки дефинираните нефункционални изисквания.
- Разработване на нови ОМ модели и тяхното симулиране с настоящия софтуерен пакет. Сред потенциалните области са изкуственият интелект, UML, описанието на бизнес процеси и много други.
- Създаване на специфични за конкретна област разширения на пакета.
- Изграждане на интегирана помощна система, която да бъде достъпна онлайн. Документацията би могла да включва както технически инструкции за работа с приложението, така и различни аспекти от теорията на ОМ и методологията за изграждане на модели.

Амбицията на автора е разработеното софтуерно приложение да добие голяма популярност в средите, занимаващи се с ОМ, да катализира разработването на нови програмни инструменти за работа с ОМ, а в комбинация с постигнатите нови научни резултати да помогне за бъдещото развитие на теорията на обобщените мрежи.

Авторска справка

Справка за приносите в дисертационния труд

Научни приноси

1. Анализирани са и са изследвани съществуващите среди за моделиране с ОМ.

2. Предложени са оптимизации на алгоритмите за функциониране на обобщени мрежи (ОМ) и някои техни разширения.

3. Дефинирани са два нови глобални оператора G'_2 и G'_6 над ОМ, а за един от първоначалните – G_6 – е предложена формална дефиниция.

4. Дефинирана е универсална обобщена мрежа, представяща функционирането и резултатите от работата на произволна мрежа на Кан (Kahn Process Networks).

Научно-приложни приноси

1. Реализирана е нова версия на симулатора за обобщени мрежи GNTicker. Новата версия използва предложения в дисертацията оптимизиран алгоритъм. Софтуерното приложение позволява конструиране на ОМ модели, използващи изчисления с MATLAB.

2. Проектирана и реализирана е Model-View-Controller-библиотека на Java, подходяща за приложения за конструиране на модели, използващи графична нотация.

3. Проектирана и реализирана е интегрирана среда за моделиране с обобщени мрежи. Средата е с разширяем модулен дизайн и удобен потребителски интерфейс.

4. Построен е и е реализиран обобщеномрежов модел за паралелно сравнение на различни математически модели на полупериодичен процес на култивация на *E. coli* с цел валидиране на разра-

ботената среда и интегрирането ѝ с MATLAB.

Публикации, свързани с дисертационния труд

[1] Dimitrov D. G. On the global operator G_2 over generalized nets. Proc. of 12th Int. Workshop on Generalized Nets, Burgas, Bulgaria, 17 June 2012, 1-5.

[2] Dimitrov D. G. On the global operator G_6 over generalized nets. Annual of "Informatics" Section, Union of Scientists in Bulgaria, Vol. 5, 2012, 43-52.

[3] Dimitrov D. G., M. Marinov. On the Representation of Kahn Process Networks by a Generalized Net. 6th IEEE International Conference on Intelligent Systems (IS'12), Sofia, Bulgaria, Sep 6-8, 2012, 168-172.

[4] Dimitrov D. G. Optimized algorithm for token transfer in generalized nets, Recent Advances in Fuzzy Sets, Intuitionistic Fuzzy Sets, Generalized Nets and Related Topics, Vol. 1., 2010, 63-68.
Към месец март 2013 г. са известни 6 цитирания.

[5] Atanassov K., D. Dimitrov and V. Atanassova, Algorithms for Tokens Transfer in the Different Types of Intuitionistic Fuzzy Generalized Nets. Cybernetics and Information Technologies, Vol. 10, 2010, No. 4, 22-35.
Към месец март 2013 г. са известни 2 цитирания.

[6] Dimitrov D., O. Roeva. Comparison of Different Mathematical Models of an E. coli Fed-batch Cultivation Process Using a Generalized Net Model. Proc. of 13th Int. Workshop on GNs, London, UK, 29 October 2012, 15-23.

[7] Dimitrov D. G. GN IDE – A Software Tool for Simulation with Generalized Nets. Proceedings of Tenth Int. Workshop on Generalized Nets. Sofia, 5 December 2009, 70-75.
Към месец март 2013 г. са известни 2 цитирания.

[8] Димитров Д. Г. Графична среда за моделиране и симулация с обобщени мрежи. Годишник на секция „Информатика“ при Съюза на учените в България, кн. 3, 2010, 51-66.

Към месец март 2013 г. са известни 2 цитирания.

[9] Димитров, Д. Г. Софтуерни продукти, реализиращи обобщените мрежи. Годишник на секция Информатика при Съюза на учените в България, кн. 3, 2010, 37-50.

Към месец март 2013 г. са известни 4 цитирания.

[10] Димитров, Д. Г. Интеграция на симулатора за обобщени мрежи GNTicker с Matlab. Годишник на секция Информатика при Съюза на учените в България, том 4, 2011, 49-52.

Към месец март 2013 г. са известни общо 16 цитирания на 5 от статиите.

Кратка биография

Димитър Димитров е роден през 1985 г. в гр. София. През 2004 г. завършва Националната природоматематическа гимназия „Акад. Л. Чакалов“. Същата година продължава образованието си във Факултета по математика и информатика на Софийския университет „Св. Климент Охридски“, специалност Информатика. От втори курс започва да води упражнения във ФМИ. През 2006 г. започва научна дейност към Българската академия на науките и скоро написва първата си публикация. През 2008 г. завършва бакалавърската степен като първенец на випуска. През 2009 г. започва работа като редовен асистент към катедра „Компютърна информатика“ на ФМИ. През 2010 г. завършва магистърска програма „Софтуерно инженерство“. Същата година е зачислен като докторант на самостоятелна подготовка към същата катедра.

Успоредно с научната работа Димитър Димитров работи и като софтуерен инженер в софтуерна фирма.

Към момента Димитър Димитров има 22 научни публикации и подготвя учебно помагало за студенти.

Научните интереси на Димитър Димитров са в областта на обобщените мрежи, размитите множества и компютърната графика.