

Софтуерни продукти, реализиращи обобщените мрежи

Димитър Г. Димитров

Централна лаборатория по биомедицинско инженерство – БАН
ул. Акад. Г. Бончев, бл. 105, София 1113, България
mitex@gbg.bg

Резюме: През годините са разработвани различни софтуерни продукти, реализиращи теорията на обобщените мрежи (ОМ) – интерпретатори на ОМ, графични редактори и т.н. В настоящата работа са разгледани всички опити и постигнати резултати за програмна реализация на ОМ и се проследява тяхното развитие. Направен е подробен и критичен хронологичен обзор на различните софтуерни продукти за работа с ОМ, разгледани са техните предимства и недостатъци. Основно внимание е обърнато на най-новия към момента пакет, носещ кодовото име GN Lite – компонентно базиран софтуер с архитектура от тип „клиент-сървър“.

1. Въведение

Обобщените мрежи (ОМ) представляват разширение на мрежите на Петри. ОМ са инструмент за моделиране и оптимизация на паралелни и конкурентни процеси в сложни системи и за решаване на задачи, за които други средства се оказват неприложими или неефективни.

В настоящата работа за първи път се прави изчерпателен исторически преглед на всички опити и постигнати резултати за програмна реализация на ОМ и се проследява тяхното развитие. През годините са започвани множество симулатори, графични редактори, дефинирани са били формати за съхранение на ОМ модели и др. За целите на настоящия обзор са изследвани всички публикации, посветени на софтуерната реализация, както и прилежащата документация към различните софтуерни пакети.

През годините са създадени стотици ОМ модели на процеси от различни области от науката и промишлеността. По отношение на медицината например от 1983 г. досега 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. Моделиращите възможности на ОМ са доказани теоретично. Създаването на софтуер, реализиращ теорията на ОМ, е важна стъпка, която дава възможност на моделиерите да се възползват от пълните възможности на ОМ.

В следващите глави се разглеждат последователно в хронологичен ред всички публикувани софтуерни решения за ОМ. Основният фокус пада върху последния пакет – GN Lite, който включва в себе си множество различни компоненти. Разгледани са предимствата и недостатъците на различните пакети.

2. Първи опити

През 1983 г., скоро след създаването на теорията на обобщените мрежи започва да се работи и върху нейната компютърна реализация на Фортран [7].

През 1987-1989 г. е написано ново софтуерно решение за обобщени мрежи. Състои се от симулатор и интерфейс за въвеждане на ОМ. Симулаторът е написан на Turbo Prolog 2.0 от Л. Атанасова, а останалата част – на Turbo Pascal от Е. Димитров.

За да извърши симулация на дадена мрежа, потребителят трябва първо да разполага с готов ОМ модел, който след това да въведе чрез потребителския интерфейс. Въведеният модел се съхранява във файл, който после ръчно се подава на симулатора.

Самото въвеждане на модел се извършва, като се въведат имена на преходи, имена на входни и изходни позиции и т.н. Въпреки че този метод е примитивен и не позволява последващо редактиране на вече въведен модел, той е удобен за бързо преобразуване на готов ОМ модел от хартиен (или идеен) вид в електронен.

Много от слабите страни на тези софтуерни пакети са наложени от тогавашните технологии, например има силни ограничения за размера на ОМ моделите (брой преходи, брой позиции и т.н.).

3. PPGN

През 1991-92 г. Р. Христов разработва Program Package for Generalized Nets (PPGN) [8] [9] [10]. Използваният език е Turbo Pascal 7.0. В този пакет основният акцент пада върху симулацията на обобщени мрежи, а потребителският интерфейс е недовършен и затова се използва този от предишния пакет.

4. STGNs

Следващият пакет се нарича Software Tools for Generalized Nets (STGNs) и е разработван през 1993 г. от Р. Христов и С. Михов [11] [12]. Пакетът е трябвало да включва следните компоненти:

- език за описание на ОМ модели;
- графичен редактор за създаване и редактиране на ОМ модели;
- инструмент за симулация на обобщени мрежи,
- инструмент за анализиране на резултатите от тяхната симулация.

Било е планирано STGNs да поддържат йерархични обобщени мрежи. Една ОМ може да съдържа множество подмрежи, които са представени в нея като преходи или позиции. Подмрежа може да се представи като позиция, само ако има един вход и изход. Когато по време на симулация дадено ядро достигне до вход на подмрежа, започва симулация на подмрежата. Симулацията на главната ОМ и нейните подмрежи се синхронизира по абсолютна времева скала. Интересно предимство на йерархичните мрежи е възможността готови ОМ модели да бъдат интегрирани в нови мрежи като техни подмрежи. Предложена е и бъдеща идея подмрежата да бъде и от друг тип, например друго разширение на мрежа на Петри (цветна мрежа и др.).

Така биха могли да се сравнят по резултат и действие модели, базирани на различни разширения на МП.

В STGNs моделите се представят със специално създаден за целта език – Language for GN Description (LGND). Той е език от високо ниво, чиито ключови думи са понятия от теорията на обобщените мрежи. С LGND се описват както структурата, така и характеристикните функции и предикатите.

Симулаторът за ОМ в STGNs позволява разцепването и сливането на ядра. Когато това е позволено, някои ядра могат да се разцепят, ако могат да преминат от една входна позиция в няколко изходни. При разцепването се получават няколко ядра с идентични характеристики.

Симулацията може да бъде както автоматична (стъпките автоматично се изпълняват последователно), така и ръчна – потребителят задава всяка стъпка ръчно. Тази възможност се среща и в следващите софтуерни пакети.

Било е планирано пакетът STGNs да въвежда удобни средства за визуализация на резултатите от симулация на ОМ модел:

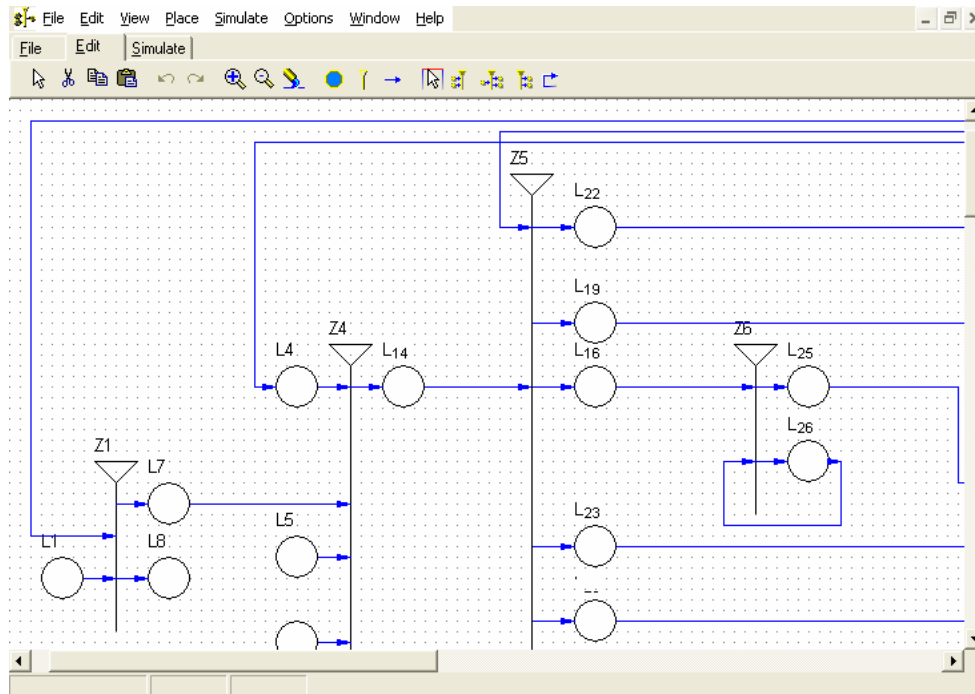
- изменението на посочени от потребителя характеристики може да се проследи чрез графична диаграма или таблица;
- за даден интервал от време може да се проследи кои преходи не са били активирани, през кои позиции не са преминали ядра и т.н.

За съжаление, по обективни и субективни причини разработката на пакета STGNs не е била завършена докрай.

5. Gennete

Gennete към момента е най-успешният софтуер за работа с ОМ. Създаден е от Х. Аладжов [13] през 1998-1999 г. Представява цялостна среда за моделиране с обобщени мрежи, която съпровожда потребителя през всички стъпки в процеса на моделирането: дефиниране на ОМ модел, стартиране и проследяване на симулация, съхраняване на модела във файл, запис на резултатите от симулацията, експортиране на модела в различни формати и много други. Софтуерът е написан на популярния за времето си Borland Delphi 5.

Първоначално Gennete е включвал и минималистичен симулатор, а впоследствие е бил използван само като графична среда към SIM2000 [6], който е споменат по-долу. По-късно, след като е бил разработен GNTicker, Gennete е бил адаптиран и към него като комуникацията между тях се извършва посредством примитивен протокол, предшественик на GNTP. GNTicker и GNTP ще бъдат разгледани детайлно по-надолу, тъй като са основна част от най-новия пакет.



Фигура 1: Gennete в режим на редактиране на модел

Предимства

- Потребителят извършва всички основни дейности по работата с ОМ чрез един софтуерен продукт.
- Удобен и интуитивен потребителски интерфейс, улесняващ дефинирането на ОМ модел и предлагащ интерактивно следене на симулацията.
- Предлага множество удобства, например автоматично построяване на дъги по зададени начало и край с минимален брой пресичания, автоматично номериране на преходи и позиции.
- Поддържа се експортиране до различни формати, най-важните от които са TeX и BMP. Това значително улеснява описването на моделите в научни статии, тъй като дефинирането на графичната структура на ОМ е безспорно по-удобно от използването на векторен редактор с общо предназначение, например вградения в Microsoft Word.
- Популярен е и се използва широко, за разлика от другите подобни решения.
- Разпространява се свободно (от март 2009 година в интернет [14]).
- Кодът е написан в обектно-ориентиран стил.

Недостатъци

- Delphi и езикът Object Pascal вече не са толкова популярни [15], официалната среда за разработка е платена и е трудно да се намерят специалисти, които да развиват проекта.
- Въпреки че потребителският интерфейс е удобен за дефиниране на графичната структура на ОМ модел, за други основни дейности понякога е объркващ. Например един ОМ модел се съхранява в до три различни файла – един за ядрата, един за функциите и един за всичко останало (което включва графичната структура).
- Използваният файлов формат за ОМ моделите е затворен, специфичен за Delphi, което е пречка в интегрирането на Gennete с други компоненти. Очаква се новата му версия да поддържа XML формата, който се използва в новия пакет. Форматът е описан по-надолу.
- Разработен е във вариант само за платформата Win32. За настоящите потребителски групи това е задоволително, но пречи за придобиването на по-голяма, особено международна, популярност.
- През годините се е налагало неколккратно да бъде „кърпен“, за да е съвместим с конкретен симулатор [6].

Интегриране в GN Lite

Gennete в момента се осъвременява с цел да стане съвместим с най-новия към момента софтуерен пакет. Използването му като наследен (legacy) графичен редактор ще е полезно докато бъде напълно завършена графичната среда с кодовото име GN IDE. Също така ще улесни експортирането на вече създадените модели, без да е необходимо да се разработва модул за импортиране на файлове, чийто формат е труден за декодиране.

6. SIM2000

През 2000 г. Н. Николов разработва симулатора за обобщени мрежи SIM2000 [18], за чийто нужди е създаден специален език за описание на модели със синтаксис, близък до математическата дефиниция на ОМ. В следващия пример е описана мрежа с два прехода – Z1 и Z2, като позиции L1 и L2 са входни за Z1, а L3 е изходна за него.

```
Z1: L1 L2 -> L3
Z2: L3 -> L4
Z1; Z2
```

Написан на Delphi, SIM2000 лесно се интегрира с Gennete. В този пакет е реализиран най-старият алгоритъм за движение на ядра в ОМ [1], но в него не е било отразено актуалното към момента състояние на теорията. По-конкретно, характеристиките се изчисляват след като всички ядра преминат от входовете към изходите на прехода, а не веднага след преминаването на конкретното ядро.

Недостатъчното бързодействие също е недостатък на SIM2000. По отношение на използвания програмен език важат същите недостатъци като при Gennete.

Адаптация на SIM2000

В [16] Д. Будакова и Л. Даковски предлагат адаптация на SIM2000. Потребителският интерфейс на приложението наподобява т.нар. „wizard“, като на всяка стъпка се въвежда определена част от модела. В началото в нея графично се визуализира даден ОМ модел, а в горната част са разположени контроли за указване на броя на изходните позиции за даден преход. След като въвеждането на преходи и позиции приключи, активират се други контроли, с чиято помощ се въвежда информация за дългите. На следващата стъпка се отваря нова форма, в която се въвеждат индексирани матрици на преходите. На последната стъпка се избират конкретни предикати, като за целта се предлага списък с готови реализирани предикати. При така конструирания интерфейс потребителят отново трябва да разполага с готов модел, преди да започне въвеждането му в приложението. Освен това има ограничения в броя на входните и изходните позиции за всеки преход.

7. GN Lite

В ЦЛБМИ-БАН и Техническият университет в Дрезден е създадено ново поколение софтуер за ОМ. Кодовото име на този пакет към момента е GN Lite. Началото е поставено през 2003 г. от Т. Трифонов и К. Георгиев с разработката на GNTicker – ефективен софтуерен интерпретатор за обобщени мрежи [5].

За разлика от предходните софтуерни продукти, текущият пакет е проектиран и реализиран в компонентно-ориентиран стил с големи възможности за надграждане на функционалността му. Дефинирани са стандарти за описание на ОМ модели, както и за комуникация с основния компонент – GNTicker. По този начин всеки компонент може да бъде разглеждан като черна кутия и заменен с друг, който спазва съответните стандарти. Отделните компоненти могат да бъдат разработени чрез различни технологии, съответно специалисти с различни умения могат да се включат в работата. Наследен софтуер може да бъде адаптиран и интегриран като компонент – напр. Gennete, както беше споменато. Създаването на нови компоненти отнема по-малко време, тъй като немалка част от функционалността се използва наготово, а различните компоненти могат да се пишат паралелно. В резултат вече съществуват много повече различни софтуерни инструменти за ОМ, отколкото в който и да е от предходните пакети.

GN Lite се отличава и с много по-напредналият стадий, до който е доведен, в сравнение с предшествениците си. Планирано е и внедряване на алгоритми с по-високо бързодействие и повече от новите аспекти от теорията на ОМ.

Следва по-подробно описание на съществуващите към момента компоненти.

GNTicker

GNTicker е централен компонент, чието предназначение е симулацията на ОМ модели. Негова отговорност е по дадена ОМ да посочи какви събития настъпват в мрежата след определен брой стъпки.

В GNTicker е реализиран ефективен вариант на най-новия алгоритъм за функциониране на ОМ. Спрямо предишните симулатори в GNTicker е отразено най-пълно текущото състояние на теорията на ОМ.

GNTicker е разработен на C++ в обектно-ориентиран стил.

XGN

Предишните софтуерни продукти за работа с ОМ използват различни специфични формати за съхранение на ОМ моделите. Много от тези формати са затворени, например използваните от Gennete, което до известна степен възпрепятства работата с тях от други софтуерни приложения за ОМ.

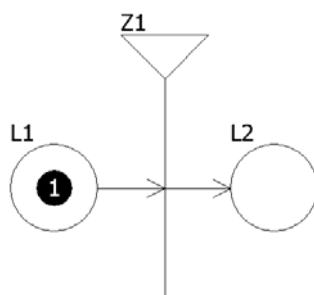
В GNTicker е дефинирана XML схема, описваща структурата на допустимите XGN файлове. Използването на XML-базиран формат за съхранение и обмен на ОМ модели има множество предимства спрямо по-ранните пакети за ОМ:

- Платформена независимост – приложения, които работят с XML файлове, могат лесно да се напишат на различни програмни езици за различни операционни системи;
- XML е изключително популярен и са създадени множество свободни инструменти за четене, запис, трансформация и обработка на XML съдържание;
- XML е разширяем. Например оригинално XGN дефинира само структурната информация за даден ОМ, но впоследствие е разширен и с визуална информация – координати на преходите, позициите и дъгите и т.н.;
- XML файловете могат да бъдат четени и с обикновен текстов редактор и лесно редактирани.

XGN файловете са организирани в четири секции – съответно за преходите, позициите, ядрата и функциите.

Секцията за функции не налага ограничения върху използвания за тях скриптов език.

Следват примерен елементарен ОМ модел и съответстващият му XGN файл [3]:



Фигура 2: Елементарен ОМ модел

```
<?xml version="1.0" ?>
<gn xmlns="http://www.clbme.bas.bg/GN" name="SimpleGN"
  time="256">
  <transitions>
    <transition id="Z1" positionX="100" positionY="100"
      sizeY="100">
      <inputs>
        <input ref="L1">
          <arc>
```

```

        <point positionX="50" positionY="150"/>
        <point positionX="100" positionY="150"/>
    </arc>
</input>
</inputs>
<outputs>
    <output ref="L2">
        <arc>
            <point positionX="100" positionY="150"/>
            <point positionX="150" positionY="150"/>
        </arc>
    </output>
</outputs>
<predicates>
    <predicate input="L1" output="L2">true</predicate>
</predicates>
</transition>
</transitions>
<places>
    <place id="L1" positionX="50" positionY="150"/>
    <place id="L2" positionX="150" positionY="150"/>
</places>
<tokens>
    <token id="alpha1" host="L1">
        <char name="Default" type="double" history="1">1</char>
    </token>
</tokens>
<functions/>
</gn>

```

GNTCFL

За GNTicker специално е разработен скриптов език за дефиниране на предикати и характеристични функции. GNTicker Characteristic Function Language (GNTCFL) значително улеснява потребителя, тъй като му дава много по-голяма мощ, отколкото например ограничено множество предварително зададени функции и не го натоварва с нуждата да пише код на C++, съобразен със специфичните особености на GNTicker. Езикът притежава синтаксис, подобен на LISP, което го прави лесен за интерпретиране. За разлика от LISP, GNTCFL не е функционален език, а процедурен. GNTicker предлага набор от примитивни процедури на GNTCFL, които включват аритметични операции, работа с масиви, достъп до характеристиките на дадено ядро, разцепване на ядра и много други.

Следва пример за предикат, написан на GNTCFL:

```

(defun W_L1_L2 "1;places.L1.obj" () (rand)
  (let (rand (random 5))
    (>= (get-len (tokens-list #0)) rand))

```

Резултатът от примерния предикат е булева стойност, която указва дали броят на ядрата в позиция L1 е по-голяма или равна на произволна генерирана стойност.

Като недостатък на GNTCFL може да се посочи неговият синтаксис, който не е особено популярен сред потребителите. Подмяната му с друг скриптов език ще

засегне единствено част от GNTicker, но няма да засегне XGN, протоколите за комуникация и много от допълнителните компоненти. По-надолу е разгледано едно решение на проблема – GNJS.

TickerServer

Разгледаният OM интерпретатор GNTicker представлява конзолно приложение. Той е удобен за напълно автоматизирана симулация на даден OM модел. Входът се чете от XGN файл, а на изхода излиза пълен списък с всички събития за всяка стъпка от симулацията на мрежата. Недостатъкът на този вариант е липсата на възможност за потребителска намеса по време на изпълнение, както и за постъпкова и графична симулация.

За да се разреши този проблем, е разработен TickerServer [6] – вариант на симулатора, комуникацията с когото се извършва посредством специален TCP/IP протокол. Предимство на избраната архитектура тип „клиент-сървър“ е възможността изпълняващият компонент (сървърът) и контролиращият компонент (клиентът) да са на отдалечени компютри с различни операционни системи. Клиентът обикновено е графична среда за моделиране с OM като Gennete [13] и GN IDE [3].

TickerServer поддържа работа с много клиенти едновременно, като симулацията на всяка OM се извършва в отделна нишка.

GNTP

GNTicker Trace Protocol (GNTP) [6] [17] е протокол за комуникация между TickerServer и неговите клиенти. Синтаксисът му е подобен на HyperText Transfer Protocol (HTTP), като целта е да може при необходимост в бъдещ момент GNTP да се превърне в разширение на HTTP.

Протоколът позволява:

- Постъпково изпълнение на мрежов модел;
- Изпълнение на мрежов модел до настъпване на дадено условие;
- Възможност за въвеждане на ядра от клиента към интерпретатора;
- Възможност за запитване за параметри от интерпретатора към клиента по време на изпълнение;
- Контрол и съобщаване на грешки.

Пример [6]:

Клиентът изпраща следната заявка към сървъра, с която поръчва да се изпълнят 10 стъпки на даден OM модел:

```
GNTP/0.1  
STEP 10
```

Сървърът изпраща следния отговор, който указва, че операцията е протекла успешно и посочва какви конкретни промени са се извършили върху модела:

```
GNTP/0.1  
200 Request completed successfully  
Content-Length: 609
```

```
<?xml version="1.0"?>
```

```

    <events xmlns = "http://www.clbme.bas.bg/GN" start="0"
end="9">
    <entrance place="L1" token="Product" step="2">
        <char name="Product" type="string">Desktop stereo
speakers</char>
        <char name="Price" type="double">15.00</char>
    </entrance>
    <movement place="L2" token="Client1" step="3">
        <char name="ID" type="double">12345</char>
        <char name="Name" type="string">John Doe</char>
    </movement>
    <exit place="L5" token="Client2" step="3">
        <char name="ID" type="double">12346</char>
        <char name="Name" type="string">Mary Jane</char>
    </exit>
</events>

```

Към момента GN IDE [3] е единственото клиентско приложение, което използва протокола GNTP за връзка с TickerServer.

GNvis

В един коректен XGN файл, описващ OM модел, не е задължително да се задава визуална информация – координати на преходи, позиции и дъги, цветове и т.н. Тя не е част от дефиницията на OM и не влияе на резултатите от симулацията. GNTicker игнорира тази информация от моделите. Такива модели обикновено не са предназначени за директна визуална симулация.

За да може графична среда за моделиране да изобрази такъв XGN файл, споменатата визуална информация трябва да бъде генерирана по някакъв начин. GNVis е създаден специално за тази цел от А. Морозов [6]. Инструментът представлява конзолно приложение, написано на езика C# за платформата .NET на Microsoft. Тъй като визуалните елементи на един OM модел могат да се подредят по различни начини, GNVis използва генетичен алгоритъм, за да създаде достатъчно удобна за възприемане структура – липса на застъпвания на преходи и позиции и минимизиране на броя на пресичанията на дъги.

GNVis може да бъде използван в среда за моделиране с OM при два различни потребителски случая:

- при отваряне на XGN файл без визуална информация;
- при нужда от автоматично повторно подреждане (auto-layout) на елементите на OM – например даден модел е бил редактиран многократно, в следствие на което са се получили множество застъпвания и пресичания на дъги.

GNJS

Едно решение на описаните проблеми на езика GNTCFL е предложено от А. Морозов и Р. Казанкин [6]. Те са разработили GNJS – разширение на GNTicker, което използва JavaScript за дефиниране на характеристични функции и предикати в OM моделите. Езикът е избран поради неговата широка популярност, лекота на изучаване и употреба, поддръжка на някои принципи на обектно-ориентираното

програмиране и наличие на множество свободни инструменти и библиотеки. GNJS може да се използва на Win32 и Unix/Linux платформи.

XGN2SVG

Както беше вече споменато, съществуват множество средства за работа с XML. Едно от тях е XSL (Extensible Stylesheet Language) [19] – фамилия от езици за трансформация и рендериране на XML документи. С тяхна помощ OM модел или част от него, записан в XML формат, може лесно да бъде преобразуван до множество текстови и XML формати, например TeX за научни статии. За целта не е необходимо да се разработва цялостно софтуерно приложение, нито дори разширение на съществуващ компонент. Достатъчно е декларативно да се напишат указания за трансформация. Този подход осигурява и платформена независимост.

XGN2SVG е разработен от А. Морозов [6]. Както подсказва името, инструментът служи за извличане на графичната структура на OM в SVG (Scalable Vector Graphics) формат. Като отворен векторен формат SVG [20] позволява лесно преоразмеряване и редактиране на изображенията без загуба на качество. Друга причина да бъде избран точно SVG формат е фактът, че той е XML-базиран и това улеснява писането на XSLT. XML2SVG би могъл да се използва и в уеб браузър за висококачествени изображения на OM модели, за разглеждането на които не се изисква инсталиране на IDE.

GNProfy

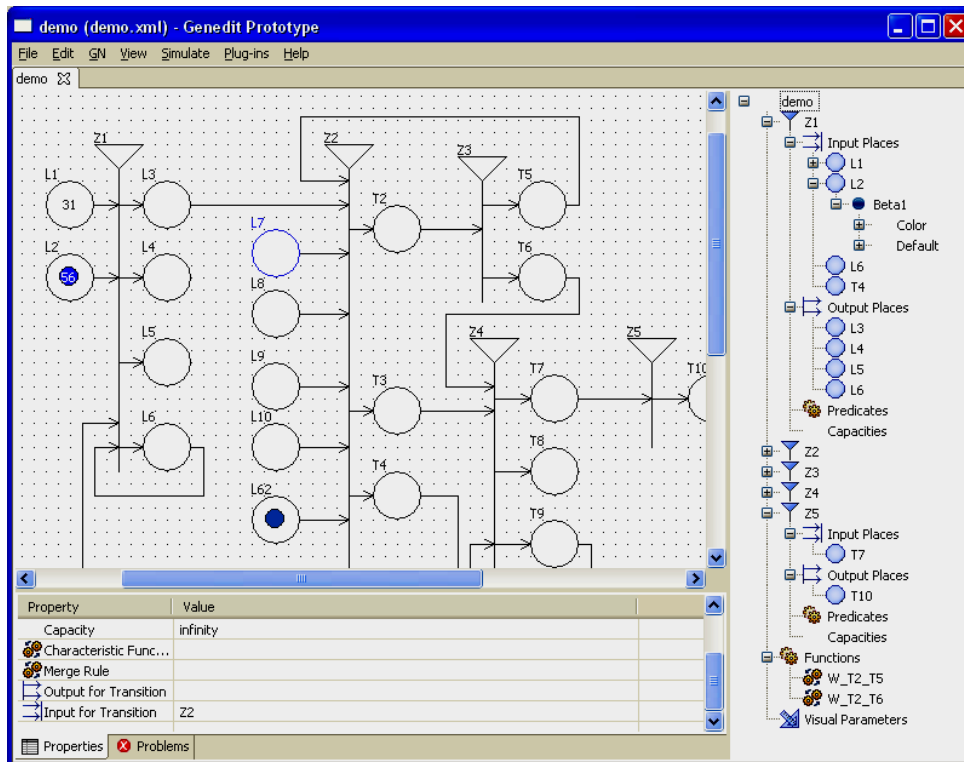
Освен че е възможно XGN файлове да се трансформират до други формати, реализиран е и обратният подход – други средства за моделиране, използващи XML файлове, да бъдат преобразувани до OM модели. С GNProfy, разработен от Е. Койчева, К. Георгиев и Т. Трифонов през 2006 г. [6] [4], е възможно UML модели, съхранени в XMI формат, да бъдат трансформирани до изпълними OM модели.

GN IDE

След GNTicker един от най-важните компоненти на софтуерния пакет за OM е графичната среда, чрез която потребителят извършва всички дейности в процеса на моделирането. Официалната графична среда към GN Lite е създадена от Д. Димитров [3]. Функционалността ѝ към момента може да се обобщи в следните точки:

- визуализиране на OM модели, съхранени в XGN файлове;
- интерактивна симулация на OM модели, използвайки външен компонент – симулационен сървър (по подразбиране – GNTicker);
- следене на параметрите на модела по време на симулация и неговата история;
- визуално редактиране на моделите и дефиниране на нови модели;
- съхраняване на моделите в XGN файлове.

За разлика от предишните подобни продукти, GN IDE може да бъде стартиран на всяка платформа – достатъчно е само да бъде инсталиран Java Runtime Environment (JRE). Реализиран е ефективен потребителски интерфейс, прилагайки утвърдени практики (фиг. 3) [3].



Фигура 3: Графична среда към GN Lite

Основен принцип на дизайна на продукта е обособяването на отделни компоненти със слаби ациклични зависимости помежду тях с възможност за повторно използване от бъдещи нови компоненти. Една от целите е да се улесни създаването на нови софтуерни компоненти, свързани с ОМ. Използвани са множество шаблони за дизайн.

Приложени са гъвкави (agile) практики при разработването на средата. Средата е разработана итеративно, в непрекъснато сътрудничество с крайните потребители.

8. Заключение

През цялото време на съществуване на теорията на ОМ е работено върху софтуерната ѝ реализация. 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 настоящия обзор е че се проследява по какъв начин развитието на теорията намира отражение в разработвания през времето софтуер.

В различните продукти 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 за обобщени мрежи – основно елементи от дефиницията на ОМ и алгоритъма за функциониране, но само малка или никаква част от другите аспекти [2] – алгебричен, операторен, топологичен и т.н.

Различните софтуерни реализации не са завършени. При всеки следващ опит е започвано почти отначало, с нов екип разработчици. Налице са различни програмни грешки. В по-старите пакети не е предвидена възможност за лесна разширяемост. Разпространението на продуктите с отворен код би помогнало за решаването на много от тези проблеми.

Основна насока в бъдещата работа е усъвършенстването на цялостната среда за моделиране с ОМ, която да е достъпна за широк кръг потребители. Съществуват нереализирани аспекти от теорията на ОМ. Средата за работа с ОМ трябва да бъде лесно разширяема, за да може да посрещне всички сегашни и бъдещи изисквания.

Благодарности

Настоящото изследване е осъществено с финансовата подкрепа на Националния фонд „Научни изследвания“ (договор ДИД-02-29).

Литература

- [1] Atanassov K. Generalized nets. World Scientific, Singapore, New Jersey, London, 1991.
- [2] Atanassov K. On generalized nets theory. Prof. Marin Drinov Academic Publishing House, Sofia, 2007, ISBN 978-954-322-237-7.
- [3] Dimitrov, D. G. GN IDE – A Software Tool for Simulation with Generalized Nets. Proc. of Tenth Int. Workshop on Generalized Nets. Sofia, 5 December 2009, 70-75.
- [4] Koycheva E., T. Trifonov, H. Aladjov. Modelling of UML sequence diagrams with generalized nets. In International IEEE Symposium on Intelligent Systems, (79 84), Varna, 2002. IEEE.
- [5] Trifonov T and K. Georgiev. GNTicker – A software tool for efficient interpretation of generalized net models. Issues in Intuitionistic Fuzzy Sets and Generalized Nets, Vol. 3. Warsaw, 2005.
- [6] Trifonov T., K. Georgiev, K. Atanassov. Software for modelling with Generalised Nets. Issues in intuitionistic fuzzy sets and generalized nets, Vol. 6, 2008, 36-42.
- [7] Atanassov K., Janev K., Atanassova L., Theory of Generalized nets (a programming aspect). Proc. of II International Symp. "Automation and Scientific Instrumentation", Varna, May 1983, 397-399.

- [8] Atanassov K., Christov R. Program Package for Generalized Nets (PPGN), Petri Net Newsletter 42, Aug. 1992, 23-26.
- [9] Mihov, S., A realization of the functions and predicates of generalized net in the simulator of PPGN for PC, First Sci. Session of the Math. Found. AI Seminar, Sofia, Oct. 10, 1989, 43-44.
- [10] Atanassov K., Christov R., About the program realization of the generalized nets, Advances in Modelling & Analysis, AMSE Press, Vol. 17, No. 1, 1993, 13-24.
- [11] Christov, R., S. Mihov, Software tools for the GNs, In: - Applications of generalized nets, (K.Atanassov, Ed.), World Scientific, Singapore, New Jersey, London, 1993, 340-348.
- [12] Mihov, S., R. Christov, On the possibility for a parallel implementation of the generalized nets, Proceedings of the 39 Internationales Wissenschaftliches Kolloquium, Ilmenau, Sept. 27-30, 1994, Band 1, 576-579.
- [13] Aladjov H., N. Nikolov, P. Georgiev, K. Atanassov. Software for generalized nets. Annual of Technical University, Sofia, Vol. 50, No. 3, 1999, 125-132 (in Bulgarian).
- [14] Gennete. Ifigenia, the wiki for intuitionistic fuzzy sets and generalized nets. <http://ifigenia.org/w/index.php?title=Gennete&oldid=1718>
- [15] Programming Language Popularity. www.langpop.com/ (last visited Dec 2009).
- [16] Budakova D., L. Dakovski. A Program System for Generalized Nets Models Constructing. International Conference on Computer Systems and Technologies – CompSysTech'06.
- [17] GNTicker Trace Protocol. Ifigenia, the wiki for intuitionistic fuzzy sets and generalized nets. http://ifigenia.org/w/index.php?title=GNTicker_Trace_Protocol&oldid=2327
- [18] Nikolov N., The SIM2000 simulation package for generalized nets: architecture and language. Issues in Intuitionistic Fuzzy Sets and Generalized Nets (K. Atanassov, J. Kacprzyk and M. Krawczak, Eds.), Wydawnictwo WSISiZ, Warszawa, 2004, 25-35.
- [19] XSLT Introduction. http://www.w3schools.com/xsl/xsl_intro.asp (last visited Dec 2009).
- [20] Scalable Vector Graphics (SVG). <http://www.w3.org/Graphics/SVG/> (last visited Dec 2009).