

ПРОТОТИПНА РЕАЛИЗАЦИЯ НА ОНЛАЙН СИСТЕМА ЗА СИМУЛАЦИЯ, ИЗПОЛЗВАЩА ВЪЗМОЖНОСТИТЕ НА МЕТОДА НА СИСТЕМНАТА ДИНАМИКА. МОДУЛИ ЗА ВИЗУАЛИЗАЦИЯ И ПРЕДСТАВЯНЕ НА ПРОЦЕСИТЕ

Павел Кюркчиев

Резюме: В ерата на високите технологии стремежа за знания и бързо развитие на големите корпорации и правителства налага инвестирането на все повече ресурси и усилия в усъвършенстване на технологиите, които биха им помогнали да осъществяват своите амбиции. Всичко това подпомага бързото развитие на изчислителните системи.

В представената статия се дава описание на един от най – известните методи за моделиране и се описва прототип на онлайн система, предоставяща възможност за моделиране с този метод.

За да бъдат задоволени всички нужни и изисквания, поставяни пред една модерна система за моделиране, тя е необходимо да следва последните тенденции в софтуерната разработка като: предоставяне на отдалечен достъп до голям брой потребители, лесно разпространение на получените резултати, гъвкавост, бързо действие и възможност за работа от не инженери.

За постигането на тези цели се наложи създаването на онлайн модулна система с гъвкав интерфейс. За работата на системата беше избран метода за моделиране на системната динамика. Моделирането на процеси в системата се осъществява от три основни модула: Модул Диаграма, Модул Графика, Модул GML (graphic model language). За представянето на математическите модели като графика са изработени пет броя графични елемента (Stock, Cloud, Variable, Flow и Influence). За работата на модула GML се наложи дефинирането на опростен език за описание на диаграми. Целта пред новия

език е да бъде максимално лесен за употреба от не инженерно насочени потребители. За целта бяха избрани (Key, Value) списъци с елементи.

За съхранението на информацията в системата се грижи външна онлайн облачна услуга, с което се улеснява достъпа до информацията след нейното пресмятане.

Ключови думи: моделиране, онлайн система, език за моделиране.

1. Увод

В последното столетие бурното развитие на високите технологии тласна към подем много промишлености и отрасли от човешкия живот. Основа на това развитие станаха системите, предоставящи милиони на брой изчисления за единица време като системите [1, 2, 4]. Основната част от тази ниша се заема от приложенията за симулация и моделиране на процеси. Съществува силна тенденция за увеличаване на техния дял, чрез разширяване обхвата им на приложение. Подробно описание на основните характеристики и тенденции при съвременните системи за моделиране могат да бъдат разгледани в следните статии [7, 8]. За да се даде отговор на това защо тези системи са изиграли и играят толкова важна роля, трябва да се направи описание на тяхната същност и на процеса им на работа.

Какво представлява моделирането?

Моделирането на процеси представлява познавателен процес, целта на който е изучаване на изследваната област. Това се постига с изграждане на моделни – приближения, отразяващи определени аспекти на обектите. Моделите са фундаментални за човешкото познание и наука. Основният начин за опознаване, използван от човешкия мозък, е чрез моделни приближения. Съществуват три основни метода използвани в моделирането и това са: агентно-ориентираното, дискретно събитийното и моделиране със системната динамика. Най-разпространената от трите типа методологии е представяне чрез системна динамика описана в [3].

2. Какво представлява метода на системна динамика и какви системи могат да го имплементират?

Системна динамика е методология и математическа техника за моделиране целта, на която е разбирането, проучването и представянето в рамка на сложни и комплексни проблеми. В днешно време методът на

системната динамика се използва за анализ и дизайн както в публичния, така и в частния сектор. Елементите на системната динамика са натрупване на *flow* в *stock* и закъснения. В методологията на системна динамика проблем или система се нарича нормална циклична диаграма (*causal loop diagram*). Нормална циклична диаграма е диаграма с проста карта на система, заедно с всичките ѝ съставни компоненти и техните взаимодействия. Чрез итерации и следствени обработки, нормалните циклични диаграми описват структурата на системата. Разбирането структурата на системата, прави възможно предвиждането на специфично поведение за определен период от време.

От всичко написано по-горе може да се изведе следното определение:

Всяка система, предоставяща възможност за математическо моделиране и представяне на процеси чрез нормални циклични диаграми, може да бъде описана като система, работеща с методите на системна динамика.

Има много известни системи, залагащи в основата си метода на системна динамика, такива са: Anylogic, Stella, както и много не комерсиални такива [5], [6], [10]. Модерните системи за моделиране (и в частност тези създадени на основата на метода на системна динамика) имат един сериозен недостатък - те предоставят услугата си за моделиране само на персонален компютър, което прави системата физически ограничена за достъп. Друг основен недостатък на подобни системи е тяхната насоченост към инженерни специалисти, което стеснява кръга на потенциалните им потребители.

Следващите няколко реда ще бъде представена една разработка, съчетаваща ДС и няколко нови методи и технологии за представяне на модели.

3. Система MSS (*Model and Simulation System*)

MSS е онлайн система, изработена на основата на *single page application*. Целта на системата е лесно представяне на процеси чрез метода на системна динамика. За постигането на целта системата разполага с три основни техники за описание на процеси, като всяка една от тях цели предоставянето на максимално опростен и улеснен интерфейс за работа от не инженерни специалисти. Модулите са:

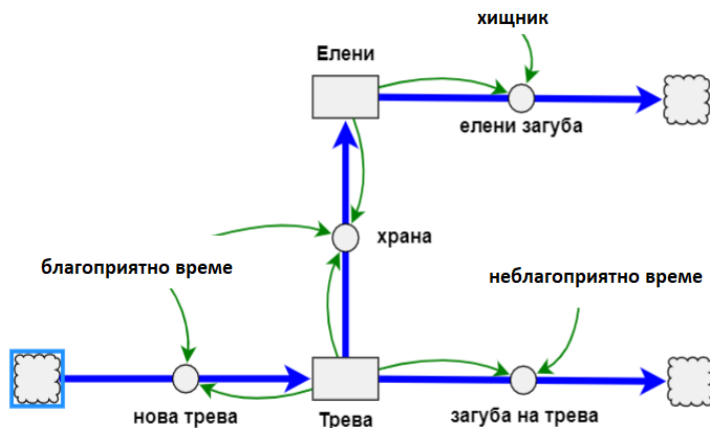
а. Модул диаграма

Модулът диаграма включва няколко основни компонента за представянето на процеси чрез нормална циклична диаграма. Това са:

- Stock е термин за субект, който се натрупва или изчерпва с течение на времето;
- Cloud е началната точка на един процес;
- Variable се използва за описание и подаване на променливи;
- Flow е степента на промяна наличността на дадения субект;
- Influence е връзката между Stock и Cloud елементите.

С комбинацията на тези елементи може да бъде изградена нормална циклична диаграма, която да моделира един комплексен проблем или процес.

Пример за такава диаграма е популацията на стадо диви елени.



На диаграмата са отразени два основни фактора, влияещи върху популацията на стадо елени. Първият фактор това са метеорологичните условия – в зависимост от това какво е времето то би повлияло индиректно върху популацията на животните. Вторият фактор е директен – това са хищниците и техните действия върху стадото.

в. Модул графика

Основната задача на модула графика е да представи данните, получени при обработката на сложните математически модели, в лесен за разбиране и следене формат. За тази цел модулет е разделен на две стандартни части. Таблична, показваща данните в табличен вид и графична. Графичната изобразява получените резултати с помощта на колонно базирана графика, което улеснява регистрирането на низходящи или възходящи трендове при обработката на модела.

с. Модул GML (graphic model language)

Преди започване реализацията на модула на „Графичен моделен език“ трябваше да се вземе решение какъв език, или схема от езици, ще бъде използвана за описание на моделите. Това постави два важни въпроса пред разработката, а именно: Използваният език трябва ли да бъде език за програмиране по своята същност, или не? Как ще бъде интерпретиран този език?

Отговорът на всеки един от тези въпроси поставя важна крачка в посока вземане на решение какво ще бъде използвано при разработката на приложението. За целта беше направен задълбочен анализ на съществуващите известни езици за програмиране и тяхното възможно внедряване. Бяха разгледани езици като Java, C# и други обвързани с парадигмите на обектно-ориентираното програмиране, както и функционални езици като F#. Най-подходящ се оказа езикът Java. Език от високо ниво, следващ парадигмите на ООП, който се използва и в доста съществуващи системи. Примерна съществуваща система с използване на Java е AnyLogic, където се използва за разширение функционалността на моделите. Всичко това е от голям плюс, но целта на приложението е да бъде максимално опростено за работа както и да удовлетворява нужни на една система, работеща по метода на системната динамика. Това прави не само Java, но и почти всички езици за програмиране, напълно неподходящи. Всеки един от описаните езици изисква специални технически знания и умения за работа. Което дава и отговора на първия по-горе зададен въпрос – използваният език не трябва да е език за програмиране, защото е нужно да бъде лесен за четене и писане от не програмисти. С други думи – той трябва да бъде интуитивен. По своята природа човешкият мозък се справя изключително добре с голяма информация, когато тя е подредена в списъци, разделени по категории. Това доведе и до решението сложните модели да бъдат разделени на категории и подредени в списъци, формирани като JSON обект. От своя страна това би улеснило работата на не програмисти с модула и изключва нуждата от създаване на сложен интерпретатор на програмен език.

Основните категории от списъци с елементи са:

- Фигури:

Списъкът съдържа всички фигури, участващи в процеса по моделиране, плюс техните координати на изобразяване върху работния плот. Името на

списъка е „nodeDataArray“. Полето „category“ описва категорията на обекта, а „label“ отговаря за името на обекта. Полетата „loc“ и „label_offset“ задават координатите на обекта;

- Връзки:

Името на списъка е „linkDataArray“. В този списък се съхраняват връзките между отделните фигури. Асоциацията на фигурите става на база полетата „category“. А полетата „from“ и „to“ описват кои обекти са свързани;

- Резултати:

Съхранява получените резултати от симулацията на математическия модел. Името на списъка е „simulationDataArray“. Полето „data“ съхранява получените стойности от измерването.

Примерен списък:

```
{
  "class": "go.GraphLinksModel",
  "linkLabelKeysProperty": "labelKeys",
  "nodeDataArray": [{
    "key": "grass",
    "category": "stock",
    "label": "Grass",
    "loc": "30 220",
    "label_offset": "0.5 0.5 0 30"
  }],
  "linkDataArray": [{
    "from": "grass",
    "to": "cloud1",
    "category": "flow",
    "labelKeys": ["grass_loss"]
  }],
  "simulationDataArray": [{
    data: [65, 59, 80, 81, 56, 55, 40],
    label: "Series A"
  }]
}
```

Системата допуска да се прави промяна в трите модула по едно и също време. За постигането на синхронизацията между тях е отговорна автономна услуга за комуникация „ConnectionService“, която следи за промени във всеки един модул. При регистрирането на промени в някой от модулите услуга

уведомява останалата част от системата на основата на observable pattern връзка.

4. Заключение

Еволюцията в софтуерното инженерство налага норми на постоянно усъвършенстване и развиване на технологиите към създаване на все по – лесно достъпни и опростени софтуерни системи. При създаването на прототипа бяха взети под внимание всички тези тенденции. Като за целта беше създаден лесен интуитивен интерфейс. След направения анализ върху възможностите за използване на програмен език като част от системата се стигна до заключението, че това няма да направи системата по - интуитивна. За решаването на този проблем се наложи създаването на нещо доста по – елементарно. Този нов език добави още един слой на работа, който направи прототипа по-достъпен за не програмисти, спрямо голяма част от съществуващите системи като AnyLogic, Simula и други некомерсиални решения като системата описана в статия [9]. Друга сериозна пречка пред развитието на прототипа беше поддържането на консистентност между различните графични модули, като това беше постигнато с изграждането на автономна услуга с роля на наблюдател, синхронизираща информацията между трите модула. Като това направи работата със системата интуитивна без да има нужда от допълнителни записи или специфични стъпки за извършване. Бъдещото развитие на прототипа ще включва имплементацията на различни не тривиални аналитични методи, като предложените в монография [11].

Благодарности

Тази работа е подпомогната от проекта ФП 17-ФМИ-008 към НПД на Пловдивски университет „Паисий Хилендарски“.

Литература

- [1] Valchanov, N., T. Terzieva, V. Shkurtov, A. Iliev, Architecture of extensible computation driven systems, Mathematics and mathematical education, *Proc. of 39th spring conference of Union of Bulgarian Mathematicians*, 06–10 April 2010, Albena, Bulgaria, 2010, 207–211.

- [2] Valchanov, N., A. Iliev, Implementation of graphical simulation environment for mathematical models, *Fundamental and Complementary Science*, “Mircea cel Batran” Naval Academy Scientific Bulletin, Constanta, Romania, Vol. XIV (2), 2011, 222–228.
- [3] Deaton M., Winebrake J., *Dynamic Modeling of Environmental Systems*, Springer, 2000.
- [4] Valchanov N., P. Petkova, A. Iliev, Integration of computational library for simulation mathematical models in web-based system for courses management, *Proceedings of the Jubilee National Conference with international participation “Tradition, directions, challenges”*, Paisii Hilendarski University of Plovdiv – branch Smolyan, 50 years Scientific and Educational Institution in the Rhodope Mountains, 19-21.10.2012, 2012, 89–94.
- [5] Илиев, А., Г. Христозов, Софтуерна среда за изграждане на динамични модели, *Proceedings of 30th International Conference „Information and Communication Technologies and Programming“* (ICT&P), София, 23–24.06.2005 г., 2005, 125–132, ISBN: 954-8986-15-9.
- [6] Илиев, А., Г. Христозов, Т. Терзиева, Софтуерна среда за представяне на динамични модели с възможност за статистика, сборник доклади от *Националната конференция „Образованието в информационното общество“*, Пловдив, 13–14.10.2006 г., 2006, 38–43, ISBN: 954-8986-22-1.
- [7] Илиев, А., Т. Терзиева, Г. Христозов, Н. Вълчанов, Анализ на съвременни софтуерни системи за динамични модели, сборник доклади от *Националната конференция „Образованието в информационното общество“*, Пловдив, 13–14 октомври 2016, 2006, 132–137, ISBN: 954-8986-22-1.
- [8] Илиев, А., Г. Христозов, Някои практически приложения на софтуерна система за представяне на динамични модели, сборник доклади от *Юбилейната конференция „Науката, образованието и времето като грижа“*, гр. Смолян, 30.11–01.12.2007 г., 64–68, 2007, ISBN: 978-954-8767-24-8.
- [9] Илиев, А., Н. Вълчанов, Т. Терзиева, Уеб базирано интерактивно помагало по математическо моделиране, сборник доклади от *Националната конференция „Образованието в информационното общество“*, Пловдив, 13–14.10.2006 г., 38–43, 2006, ISBN: 954-8986-22-1.

- [10] Илиев, А., Имитационно моделиране. Изграждане на софтуерна среда за представяне на детерминирани динамични модели, Университетско издателство „Паисий Хилендарски“, Пловдив, 2011, ISBN: 978-954-423-713-4.
- [11] Iliev, A., N. Kyurkchiev, *Nontrivial Methods in Numerical Analysis: Selected Topics in Numerical Analysis*, LAP LAMBERT Academic Publishing, Saarbrücken, 2010, ISBN: 978-3-8433-6793-6.

Факултет по математика и информатика
Пловдивски университет
Бул. „България“ № 236,
Пловдив 4003, България
Емайл адрес: pavkyu@gmail.com

PROTOTYPIC REALIZATION OF ONLINE SIMULATION SYSTEM USING THE DYNAMIC SYSTEMS METHOD. MODULES FOR VISUALIZATION AND PRESENTATION OF PROCESSES

Pavel Kyurkchiev

Abstract: In the era of advanced technologies, the urge for knowledge and fast development of big corporations and governments is forcing them to investment more resources and work into the advancement of science, which in turn can aid them in the realization of their ambitions. This whole situation leads to the fast growth of areas like computational systems.

In the following article a description of the most renown methods of modelling will be presented as well as a prototype of an online system that incorporates such methods.

To fulfil all the needs and requirements of a modern modelling system, the latest trends in software development were used. Trends like: remote access for big number of users, ease of distribution of final results, flexibility, quick response and the ability for access for non-technical users.

For the accomplishment of such tasks the creation of an online module system with flexible interface had to be completed in advance. The working method of the system is the method of modelling of system dynamics. The process modelling in a system was accomplished using three main modules: Diagram, Graphics and GML (graphic model language). For the delivery of mathematical modules as graphics five more elements have been created (Stock, Cloud, Variable, Flow and Influence). For the work of the GML module, the addition of a simplified language for describing diagrams was added. The aim of the new language was to present an easier to use method for non-technical users. Special key value list of elements were used to accomplish this goal.

A separate cloud service has been used for the storage of information in the system, which allows the ease of access to the information after it has been processed.