

Потребителски интерфейс на редактор-генератор на структурни схеми

Искра Бонева, Полина Атанасова, Ирина Желязкова

The User Interface of an Editor-Generator of Structural Schemes: The work is the second of a sequence of papers concerning an editor-generator of structural schemes with a script language. The present paper focuses on the user interface that is easy for use, intuitive for understanding and flexible for resource reusing.

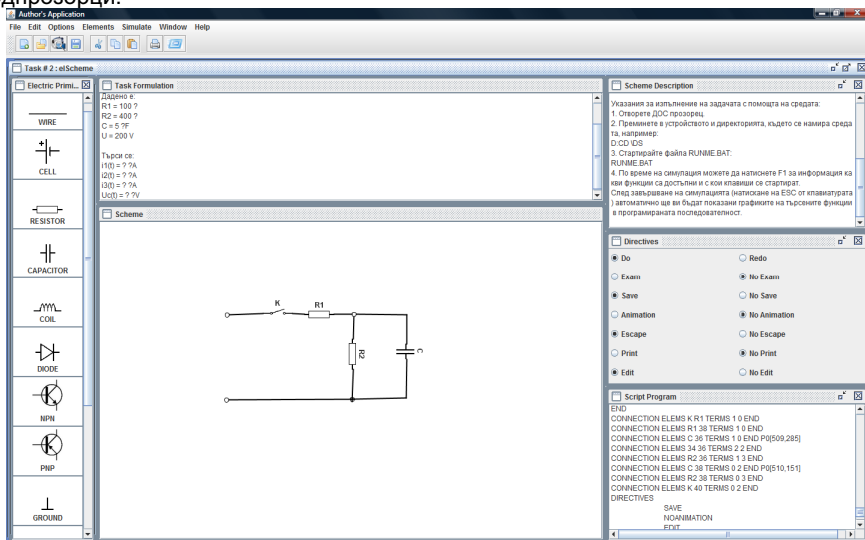
Key words: User Interface, Author's Tool, Editing and Generating, Structural Schemes.

ВЪВЕДЕНИЕ

В [5] е представена цялостната концепция на колектива за задачно-ориентирана среда (ЗОС) за построяване на различни видове структурни схеми (СС). UML проект на WINDOWS-базиран прототип на ЗОС е докладван в предишни работи [3,4]. Авторското средство за подготовка на задачите в нея е наречено редактор-генератор на програми (РГП). С помощта му авторът (А) може да: подготвя отделни индивидуални задачи, дава препоръки за намеса на преподавателя (П) по време на изпълнение на задачите от обучавания (О), както и да показва контекстно-зависим лекционен материал. За по-продуктивна работа А трябва да е запознат със: заложения основен принцип, скриптовия език **SCHEMEScript** за описание на структурните знания [2], както и с потребителския интерфейс. Настоящата работа фокусира върху потребителския интерфейс на нов JAVA вариант на РГП като последователно представя: основният прозорец, създаването на задача, решаването ѝ, както и две примерни схеми. Заключениеето съдържа основните резултати и бъдещи планове на колектива.

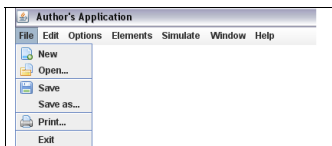
ОСНОВЕН ПРОЗОРЕЦ

След стартиране на изпълнимия файл се появява празен основен прозорец с подпрозорци.

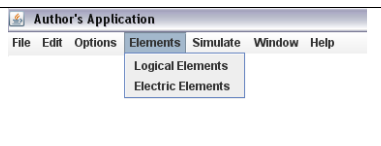


Фиг. 1. Основният прозорец на средството

На фиг. 1 те са запълнени. Най-отгоре са: главното меню и лентата с инструменти. Отношение към РГП имат само два меню-елемента (**File** и **Elements**). От подменюто на първия (фиг. 2), А може да избира измежду следните опции: създаване на нова задача, отваряне на съществуваща, запазване, принтиране и изход от приложението. От подменюто **Elements** (фиг. 3) той може да сменя типа на елементите (засега само логически и електрически). Изборът води до промяна на списъка с графичните примитиви в подпрозореца отляво (**Logical/Electric Primitive**).



Фиг. 2. Подменю File



Фиг. 3. Подменю Elements

Лентата с инструменти осигурява по-бърз достъп до някои често използвани функции (фиг. 4). Предназначението на отделните бутони с подсказки е:



Фиг. 4. Лентата с инструменти

-  - създаване на нова задача (**Create new task**);
-  - отваряне на съществуваща задача от скриптов файл (**Open task from script**);
-  - отваряне на диалоговия прозорец за сравнение на две схеми (**Open compare dialog**);
-  - запазване на растерно изображение и скриптов файл (**Save task as image and script**);
-  - обновяване на скрипта на задачата (**Regenerate script**).

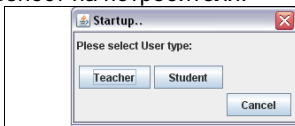
В рамката на основния прозорец се изписва: **Task #** <пореден номер>: [<име на схемата>]; където <пореден номер> е номерът на поредния създаден прозорец при текущото стартиране на средството; <име на схемата> – въведено от А име на схемата по време на създаване на основния прозорец. Ако не е въведено име, нищо не се изписва. В основния прозорец се съдържат следните подпрозорци:

- **Task Formulation** – подпрозорецът, разположен в средата отгоре. В него в свободен, но прецизен текст А въвежда формулировката на задачата;
- **Scheme Description** – подпрозорецът, разположен отдясно и отгоре. В него също в свободен текст се пишат обяснения, указания или друга допълнителна информация, свързана със задачата;
- **Directives** – подпрозорецът под **Scheme Description** позволява да се задават правата и ограниченията на обучавания (О), който ще решава задачата под мониторинга на П. Отделните групи радио-бутони имат подразбиращи се значения на разрешение или забрана. Например, **Do|Redo** за повторно изпълнение на задачата, **Exam|Noexam** - оценка степента на близост на решението на О до това на А; **Edit|Noedit** - редактирането й, **Escape|Noescape** - отказ от изпълнението й; **Save|Nosave** - съхраняването му във дисков файл; **Print|Noprint** - принтиране на файла с решението; **Help|Nohelp** - достъп до съответния лекционен материал; **Animation|Noanimation** – постъпково изпълнение на задачата.
- **Logical/Electric Primitive** – подпрозорецът, разположен отляво във вид на панел. В него се сменят графичните примитиви за типовете елементи.
- **Scheme** – подпрозорецът за изчертаване на схемата (чертожната област).

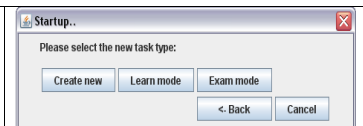
- **Script Program** – подпрозорецът долу отдясно. В него се генерира скриптът за описание на изчертаваната схема на езика **SCHEMEScript** [5].

СЪЗДАВАНЕ НА ЗАДАЧА

Създаването на нова задача може да стане по два начина. Единият е чрез натискане на бутона  от лентата с инструменти, а вторият с меню командата **File | New**. И при двата варианта се появява диалоговият прозорец от фиг. 5, който служи за избор на типа на потребителя, т.е. А или О (**Teacher/Student**). Тук и по-нататък за отказ от създаването на задачата се използва командният бутон **Cancel**. След избор на потребител в диалоговия прозорец (фиг. 6) се изисква отвърждаване на избора за създаване на нова задача (**Create new**) поради продължителността на този процес и необходимостта от предварителна подготовка на потребителя.

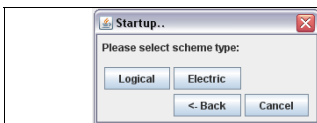


Фиг. 5. Избор на потребител

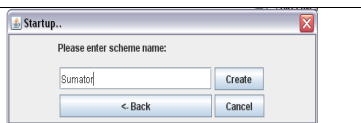


Фиг. 6. Избор за нова задача

Тук и по-нататък **Back** връща една стъпка назад (в случая към избор на нов потребител). Следва избор на режим на работа със схемата: представяне на авторското решение (**Learn mode**) с останалата допълнителна информация или построяване на собствено решение от О с цел диагностика и оценяване (**Exam mode**). Следва да се избере типа на схемата (фиг. 7) – логическа или електрическа. Ако потребителят е изпитван О, той трябва да зададе и свое собствено име на схемата (фиг. 8). Натискането на бутона **Create** връща към основния прозорец (фиг. 1).



Фиг. 7. Избор за тип на схемата



Фиг. 8. Въвеждане име на схемата

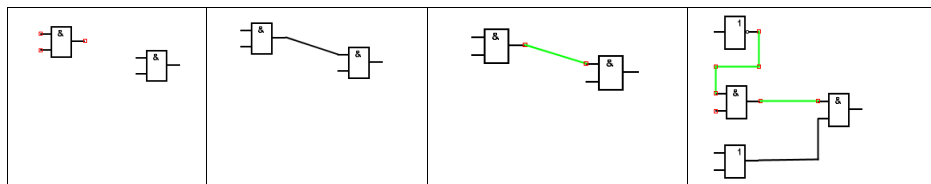
ПОСТРОЯВАНЕ НА СХЕМА

Построяването на схемното решение включва: поставяне на елементи върху чертожната област **Scheme**, свързването им чрез примитива **Wire** и последващи обработващи операции. Поставянето на елемент става чрез кликане с левия бутон на мишката върху желанния елемент в подпрозореца **Logical/Electric Primitive**. Следва втори клик за позициониране в **Scheme**.

Когато се посочи елемент с мишката в това поле, изводите му се оцветяват в червено (фиг. 9). За свързване на два елемента първо се избира примитивът **Wire**, след това се посочва изводът на единия елемент. Появата на червено квадратче означава, че е възможно линията да се свърже с елемента. Кликва се веднъж върху квадратчето и след това избира изводът на другия елемент. При кликане върху линия, свързваща два елемента, тя се оцветява в зелено. Този начин позволява да се види откъде минава дадена линия и кои елементи тя свързва. Друг начин за проследяване на връзките е, като се посочи с мишката един елемент, при което се оцветяват всички линии, свързани с него.

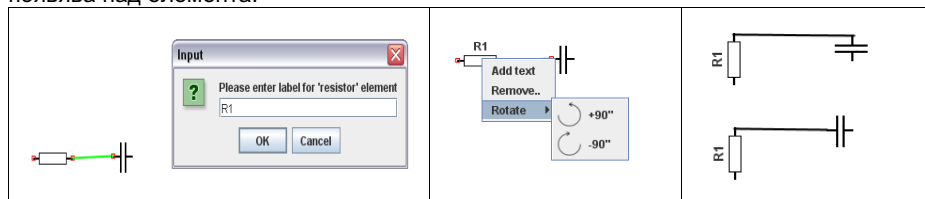
Особеност на примитива **Wire** е, че има точка на пречупване в чертожната област (фиг.10). Тя се добавя с двойно кликане на мишката върху линията. В мястото, където е добавена, даденият сегмент от линията се разделя на два самостоятелни сегмента за преместване. Когато се кликне и задържи върху точката

на пречупване, линията може да се мести. Когато се посочи линия, виждат се и точките на пречупване.



Фиг. 9. Процесът на свързване на елементи

При избор на елемент с десен бутон на мишката (фиг.10) се появява контекстно меню за добавяне на текст към елемента, за премахване на избрания елемент и за завъртане на елемента с надписа в двете посоки. При избор за добавяне на надпис се появява допълнителен прозорец за въвеждане на желанния текст като има възможност за отказ с **Cancel**. При потвърждаване с **OK** въведеният текст се появява над елемента.



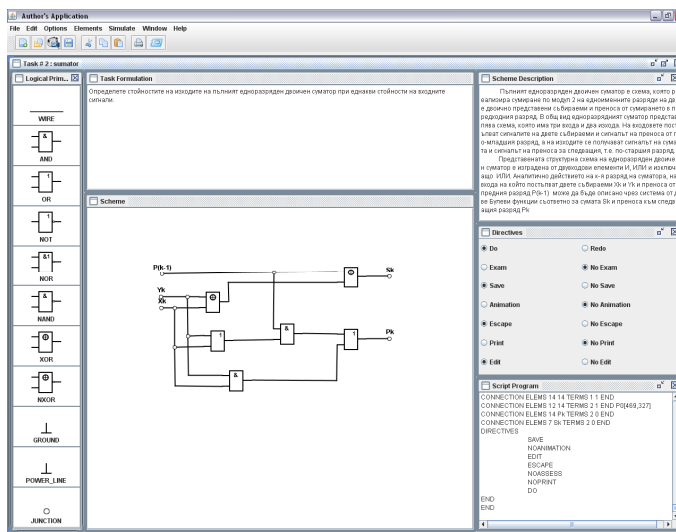
Фиг. 10. Процесът на построяване на схема

При избор на операцията ротация се появява подменю за избор на посоката на завъртане със стъпка +90 или -90 градуса (фиг.10). На фигурата са показани два резултата от изпълнение на тази операция. При местене на елемент с мишката, отместването става със стъпка 10. За по-финно преместване на елемента се задържа клавишът **Ctrl**. При изтриване на елемент се изтриват едновременно и надписът, и връзките му.

ПРИМЕРНИ ЗАДАЧИ

Завършената функционалност на средството се демонстрира с две примерни схеми. На фиг. 1 е показан резултатът от изпълнението на задача за построяване на електрическа схема от дискретни елементи. Последователно са свързани ключ и резистор, паралелно са свързани кондензатор и друг резистор. Оформилите се две групи са последователно свързани към захранващо напрежение.

На фиг. 11 е показан резултатът от изпълнението на задача за построяване на пълен едноразряден двуичен суматор за k -я разряд. Както е известно, такава комбинационна схема има три входни сигнала съответно за двете събираеми (x_k и y_k) и преноса от предишния разряд p_{k-1} и два изходни сигнала съответно за сумата (s_k) и преноса от k -я разряд (p_k). В случая схемата е построена от три вида логически елементи: сума по модул 2 (2 броя), "И" (два броя) и "ИЛИ" (два броя).



Фиг. 11. Задача за построяване на комбинационна схема

ЗАКЛЮЧЕНИЕ

Представен е лесен, удобен и гъвкав потребителски интерфейс на средство за автора/обучавания за подготовка/изпълнение на задачи за построяване на структурни схеми. UML проектът на този нов JAVA реализиран вариант е даден в [1]. Средството е независимо от типа на схемите и генерира подпрограми на скриптов език за описание на задачите. То е отделено от текстовия редактор за подготовка на програми за упражнения от преподавателя в задачно-ориентирана среда.

ЛИТЕРАТУРА

- [1] Бонева Й., Атанасова П., Желязкова И. UML проект на средство за подготовка на задачи за структурни схеми, Трудове на научната конференция на РУ, 2009.
- [2] Zheliazkova I. I., Georgiev G. T., Valkova P. L. A Visual Language for Structural Knowledge Description, *Newsletter of the Union of Scientists – Rousse, Section of Mathematics, Informatics, and Physics*, Vol. 5, 2004, pp. 111-120.
- [3] Atanasova P. L., Stefanov S., Zheliazkova I. I. An Adaptive and intelligent Text Editor for Teacher's Support of Practical Exercises, *Proceedings of International Scientific Conference Computer Science'2008*, Sept. 18th-19th, Kavala, Greece, part II, pp.456-461
- [4] Zheliazkova I. I., Georgiev G. T., Valkova P. L. An Editor-Generator of Programs for Structural Knowledge Description, *International Conference on Computer Science'5*, Chalkidiki, Greece, 30 September-2 November, 2005, part II, pp. 188-193.
- [5] Zheliazkova I. I., Georgiev G. T., Valkova P. L., A Task-Oriented Environment for Structural Schemes Design, *Information Technologies and Control*, Vol. 2, 2006, pp. 2-13.

За контакти:

Доц. д-р Ирина Желязкова, Катедра "Компютърни системи и технологии", Русенски университет "Ангел Кънчев", тел.: 082-888 711, e-mail: Irina@ecs.ru.acad.bg

Докладът е рецензиран.