

## ВЗАИМОДЕЙСТВИЕ СИСТЕМЫ КОНСТРУИРОВАНИЯ УЧЕБНЫХ ТЕСТОВ С ПОДСИСТЕМОЙ РЕШЕНИЯ ГРАФИЧЕСКИХ ЗАДАЧ

Кундро А.В.<sup>1</sup>, Полозков Ю.В.<sup>1</sup>, Полозков А.Ю.<sup>2</sup>

1) Белорусский национальный технический университет

2) Белорусский государственный университет  
информатики и радиоэлектроники  
Минск, Республика Беларусь

В целях развития онлайн-образования разрабатывается клиент-серверное web-приложение «Онлайн редактор для решения тестовых графических задач», включающее три основных модуля, представленных собственным интерфейсом. Первый модуль обеспечивает компоновку элементов, выступающих связующим звеном для дальнейшей разработки алгоритмов в соответствии с областью их применения. Второй представляет собой конструктор алгоритмов решения задач, позволяющий задавать и редактировать описательные условия, входные параметры и исходные данные учебной задачи, используя вышеописанные элементы. Этот конструктор позволяет формировать последовательность алгоритмов для контроля составляющих этапов решения общей задачи, а также связывать входные факторы и выходные параметры. Третий модуль предоставляет возможность взаимодействия пользователя с созданными экземплярами задач для их выполнения. Пример разработанной последовательности алгоритмов для графической задачи по построению профильной проекции пространственной точки приведен на рисунке 1.

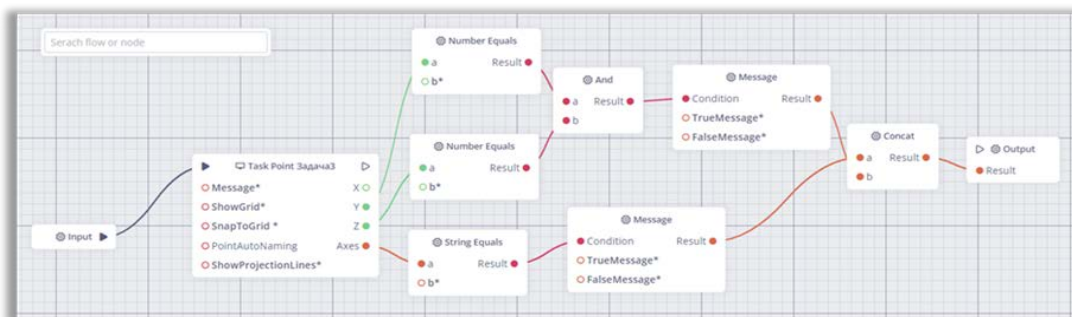


Рисунок 1 – Пример формирования алгоритма контроля выполнения графической задачи

Решение учебных задач выполняется обучающимся в графическом окне в соответствии с заданными описательными условиями и корректируется путем изменения значений контролируемых (входных) факторов (рисунок 2).

Приложение реализовано с использованием фреймворков .NET Core, Entity Framework Core и React.js. Пользовательский интерфейс построен с применением библиотек Bootstrap, MUI и Reactstrap.



Рисунок 2 – Пример выполнения задачи в графическом окне приложения

Для решения задачи в графическом интерфейсе в коде указывается специальная команда. Страница внешнего графического конструктора отображается внутри iframe web-приложения. Связь между iframe и web-приложением осуществляется через обработчик события «message» и метод send. Это позволяет отправлять параметры во внешний интерфейс и получать результат после выполнения определённых действий. После получения результата происходит переход к следующему узлу, который выполняет свою команду, получая определенный результат. Важным моментом является то, что выполняются только те узлы, у которых известны все параметры. Если какой-либо параметр неизвестен, узел ожидает пока он заполнится. Для работы графического интерфейса, программный код, обрабатывающий команду, реализован на стороне внешнего конструктора:

```

window.splate = {
  url: window.location !== window.parent.location
    ? document.referrer
    : document.location.href,
  send: (data) => { window.parent.postMessage(data, splate.url); },
  onSubmit: () => { },
  onReceived: (data) => { }
};
window.addEventListener("message", e => {
  if (e.data === "submit") {
    splate.onSubmit();
  }
  else {splate.onReceived(e.data);}
});

```

Для функционирования обработчика переопределяются функции onSubmit и onReceived. Функция onSubmit срабатывает при нажатии на кнопку «Submit». При этом необходимо вызвать функцию send и передать объект с ответом, который затем преобразуется в выходные параметры узла. Функция onReceived выполняется при запуске узла, а в качестве параметра функции передаются входные данные узла. Если после вызова onSubmit функция send не отвечает в течение 10 секунд, узел отклонит обработку.