

РАЗРАБОТКА КОМПЬЮТЕРНОГО ИМИТАЦИОННОГО ТРЕНАЖЕРА «КУСТОВАЯ ПЛОЩАДКА СКВАЖИН»

Аннотация. В статье представлены этапы разработки компьютерного имитационного тренажёра «кустовая площадка скважин»

Ключевые слова: имитационное моделирование, оптимизация распределенные информационные системы, информационные системы, тренажеры компьютерные, тренажеры моделирующие, тренажерно-моделирующие комплексы, программное обеспечение.

Современное производство характеризуется увеличивающимися темпами внедрения передовых научных, технических, организационных и экономических разработок. Постоянное увеличение доли высокотехнологичного оборудования в различных секторах производства, вместе с увеличением сложности оборудования и производственных процессов в целом, закономерно ставит задачи повышения качества тренинга специалистов работе с оборудованием. Тренинга, позволяющего специалисту принимать адекватные решения при штатных, нештатных и аварийных ситуациях, возникающих на производстве. Качество подготовки специалистов в значительной степени определяет экономическую эффективность производства (напрямую зависит от эффективности действий персонала), а также затрагивает вопросы охраны труда, промышленной и экологической безопасности.

Согласно отчетам о деятельности Федеральной службы по экологическому, технологическому и атомному надзору (Ростехнадзор), большое количество аварий и несчастных случаев в различных секторах производства вызвано не только старением основных фондов предприятий, но и человеческим фактором. Наиболее характерными причинами, согласно отчетам комиссий Ростехнадзора, являются:

- низкая технологическая дисциплина;
- неосторожные или несанкционированные действия при выполнении работ;
- слабые знания персонала относительно требований безопасности ведения работ;
- недостаточная подготовленность (психологическая и квалификационная) персонала;
- недостаточная эффективность обучения и инструктажа персонала по вопросам безопасности;
- несогласованные и ошибочные действия персонала в условиях чрезвычайной ситуации;
- неправильные действия обслуживающего персонала по ведению технологического процесса;
- ошибки персонала при проведении необходимых измерений;
- несвоевременное обнаружение предаварийной ситуации при наличии характерных признаков;
- несвоевременное принятие мер по устранению аварии;
- нарушение технологии и регламента выполнения работ;
- не предусмотрены мероприятия для ликвидации возможных аварий;
- ненадлежащее проведение инструктажа и т. д.

Помимо аварий, неверные действия обслуживающего персонала часто приводят к внеплановым остановам технологического процесса, что также является источником значительных экономических потерь предприятий.

Применение компьютерных имитаторов не требует вмешательства в производственные процессы, исключает в процессе имитации возможность возникновения реальных инцидентов и аварий в случае неверных технических и технологических решений и ошибочных действий персонала. Материальные затраты при компьютерной имитации значительно меньше, чем при выполнении аналогичных работ на реальном оборудовании. [1]

Эти факты являются одним из обстоятельств, который подчёркивает актуальность моей темы. Целью разрабатываемого тренажёра является повышение доступности кустовой площадки для отработки навыков работы, на месторождении реализованных в тренажёре. Рассмотрим задачи, которые нужно решить для достижения поставленной цели.

Задачи:

1. Создание математической модели технологических процессов;
2. Создание 3D изображений и звуков для создания условий, аутентичной площадки;
3. Создание имитации интерфейса для управления объектами автоматизации и технологическими процессами (АСУ ТП);
4. Создание модуля распределённых вычислений и симуляции для актуализации всех представлений модели;
5. Создание программной реализации модуля инструктора;
6. Программная реализация математической модели технологических процессов;

Обзор составных частей тренажёра

Рассмотрим модули, приложения и оборудование, которые участвуют в разработке и работе обучающего тренажера.

Оборудование:

1. Станок-качалка;
2. Скважина, добывающая УЭЦН;
3. Скважина, эксплуатируемая фонтанным способом;
4. Нагнетательная скважина;
5. Автоматизированная групповая замерная установка;
6. Площадка насосного агрегата типа ЦНС;
7. Сосуд работающий под давлением;
8. Блок реагентного хозяйства.

Структура многопользовательского тренажера:

Модуль синтеза 3D изображения [5] и звука [4]: Компьютерная программа, реализующая процессы передачи на основные органы восприятия пользователя программно управляемых воздействий, а также процессы получения производимых пользователем действий для последующего обеспечения реалистичной реакции имитируемой среды.



Рис. 1 3D модель кустовой площадки

Модуль имитации АСУТП: Достоверно воспроизводит интерфейс реальной системы управления объектами автоматизации и технологическими процессами. Работа с модулем имитации АСУТП выполняется аналогично реальной системе АСУТП.

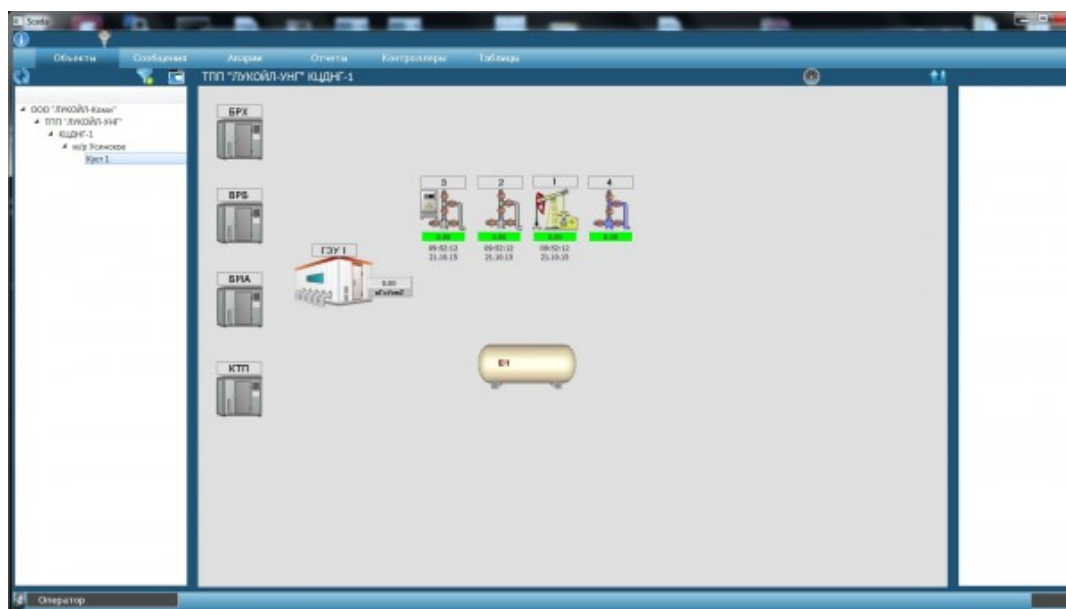


Рис. 2 Модуль имитации АСУТП

Модуль математической модели: Компьютерная программа, реализующая систему математических соотношений, описывающих с требуемой точностью

имитируемые объекты и процессы. Обеспечивает многопользовательский доступ к тренажеру и возможность интеграции тренажера в распределенную тренажерную систему, а также хранение и предоставление сохраненных сценариев.

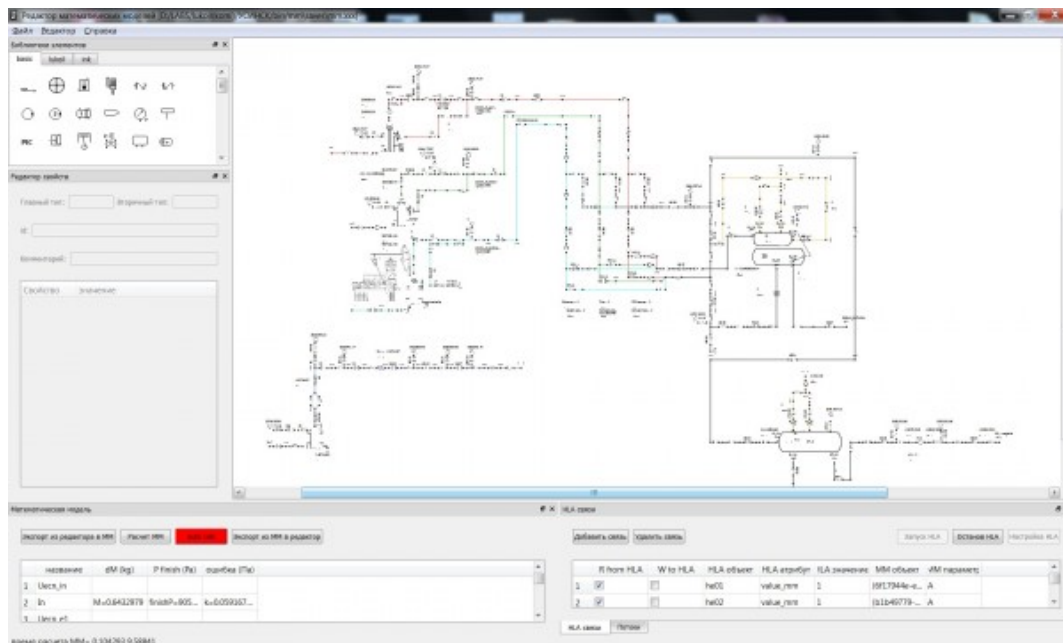


Рис. 3 Модуль математической модели

Модуль синхронизации с учебным полигоном: Компьютерная программа, обеспечивающая взаимосвязь модуля синтеза 3D изображения и звука с оборудованием учебного полигона.



Рис. 4 Фотография учебного полигона.

Модуль инструктора: Компьютерная программа, реализующая интерфейс инструктора, с помощью которого осуществляется выбор производственных ситуаций, задач и параметров оборудования.

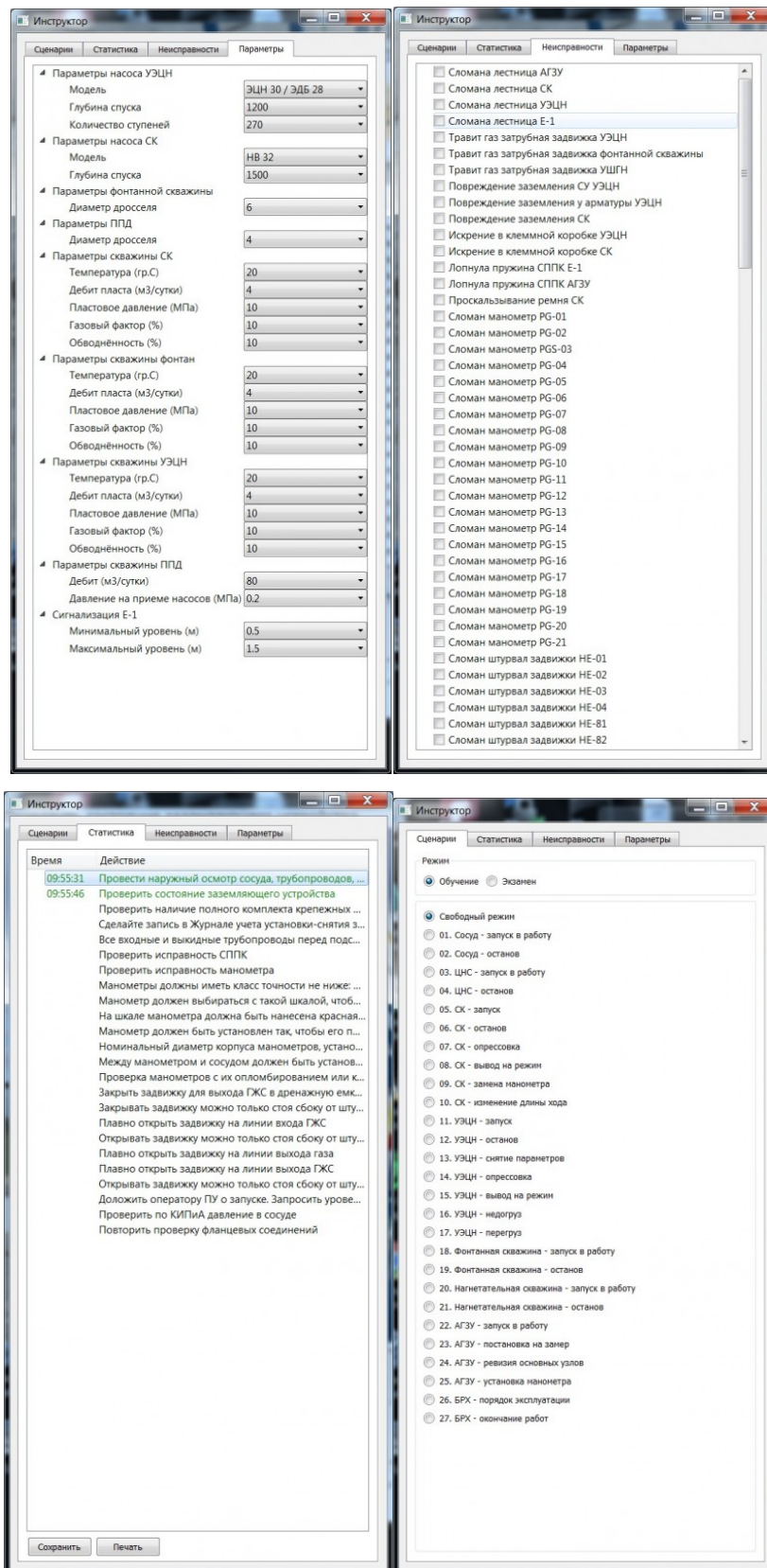


Рис. 5-8 Модуль инструктора

Редактор сценариев.

Компьютерная программа, позволяющая создавать сценарии для обучения.

Ранее реализация сценариев выполнялась путём написания скриптов на языке LUA [6], было решено оптимизировать этот процесс написав программу для создания обучающих сценариев.

Обзор редактора сценариев:

Данная программа позволяет составлять сценарии, не прибегая к языкам программирования, что ускоряет создание сценариев.

Она состоит из элементов:

1. HLA links.

Обеспечивает обращение данной программы с другими элементами тренажёра в системе федератов.

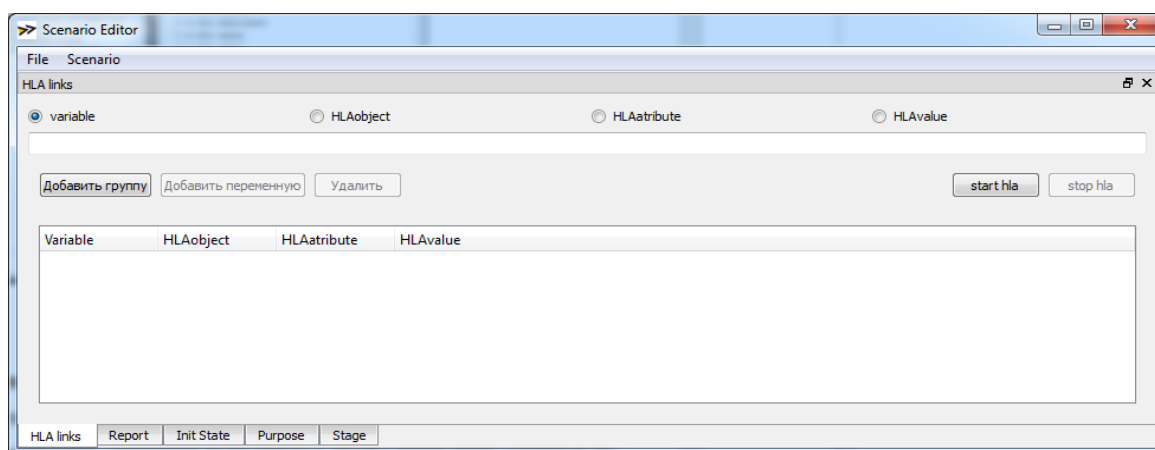


Рис. 9 Редактор сценариев вкладка HLA links

2. Report.

Отображает результаты выполнения сценария.

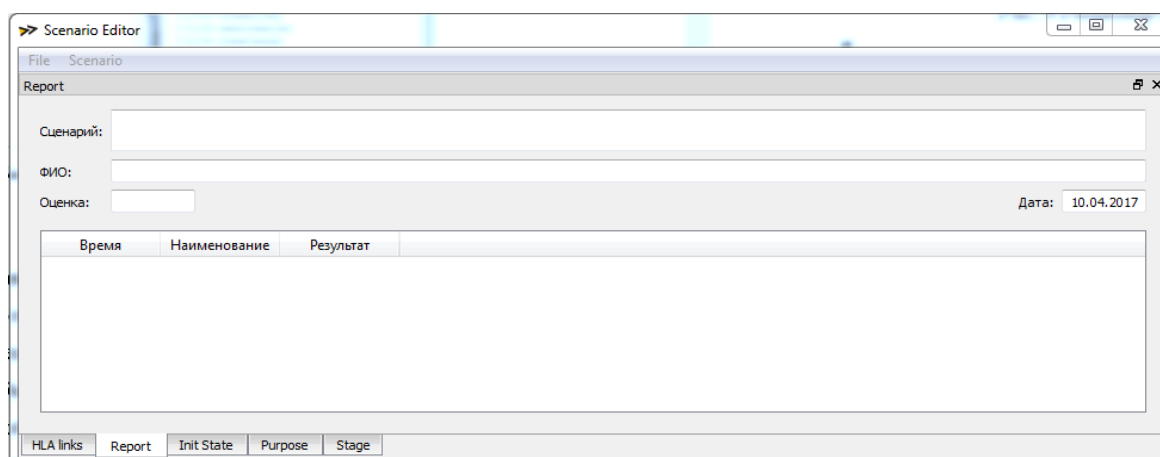


Рис. 10 Редактор сценариев вкладка Report

3. Init State.

Связывает конкретный сценарий с определённой математической моделью объекта, задаёт стартовые координаты пользователя на площадке, выводит на экран вводные инструкции по выполняемому сценарию.

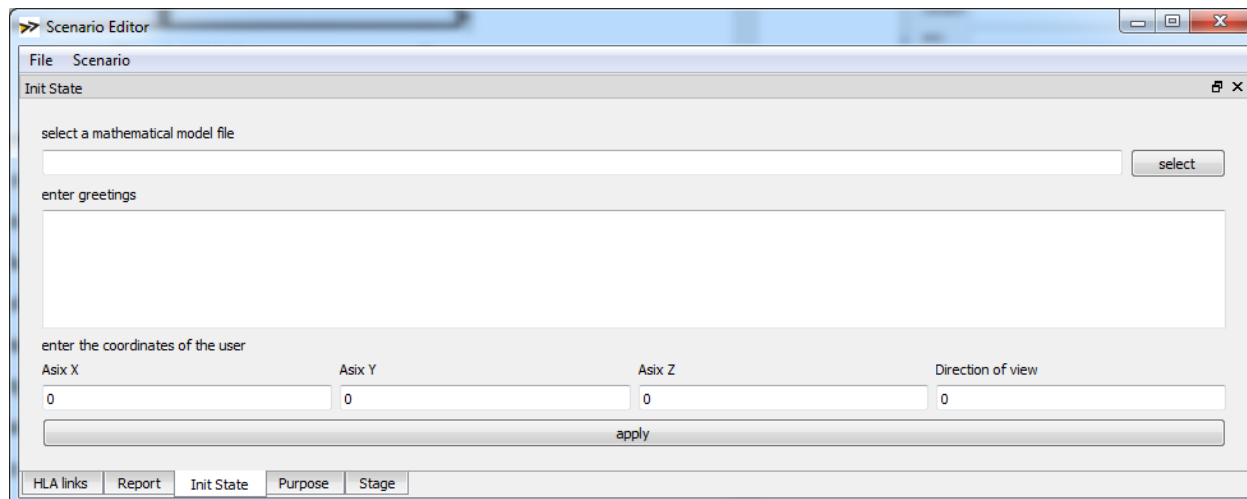


Рис. 11 Редактор сценариев вкладка Init State

4. Purpose.

Данная вкладка содержит: глобальные цели, при выполнении которых сценарий завершается и условия прохождения сценария, при нарушении условий сценарий останавливается.

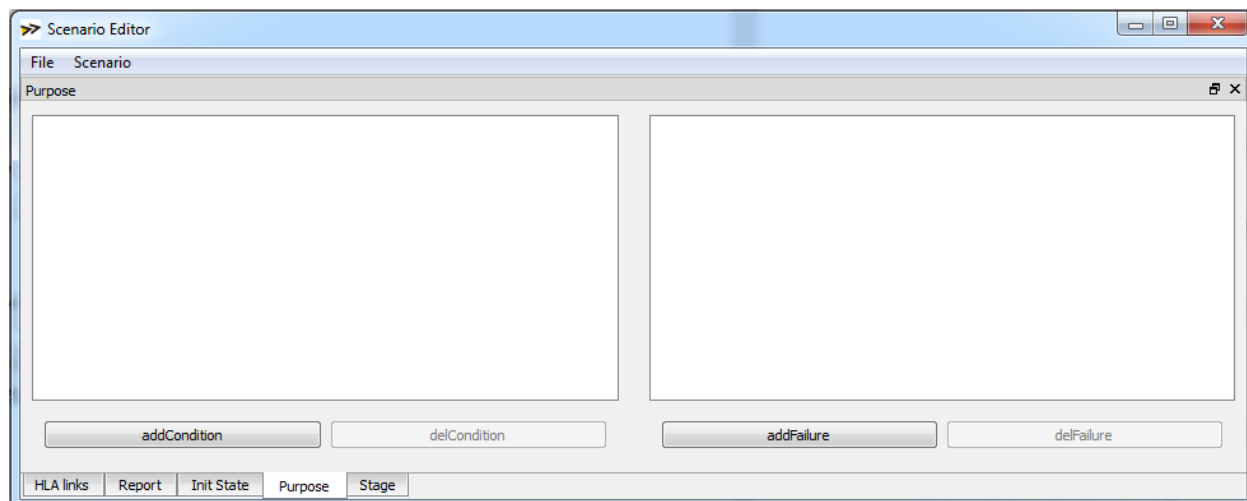


Рис. 12 Редактор сценариев вкладка Purpose

5. Stage.

Данная вкладка позволяет добавлять и редактировать шаги сценария, задавать их последовательность, выбирать их тип и добавлять условия прохождения определённой стадии сценария.

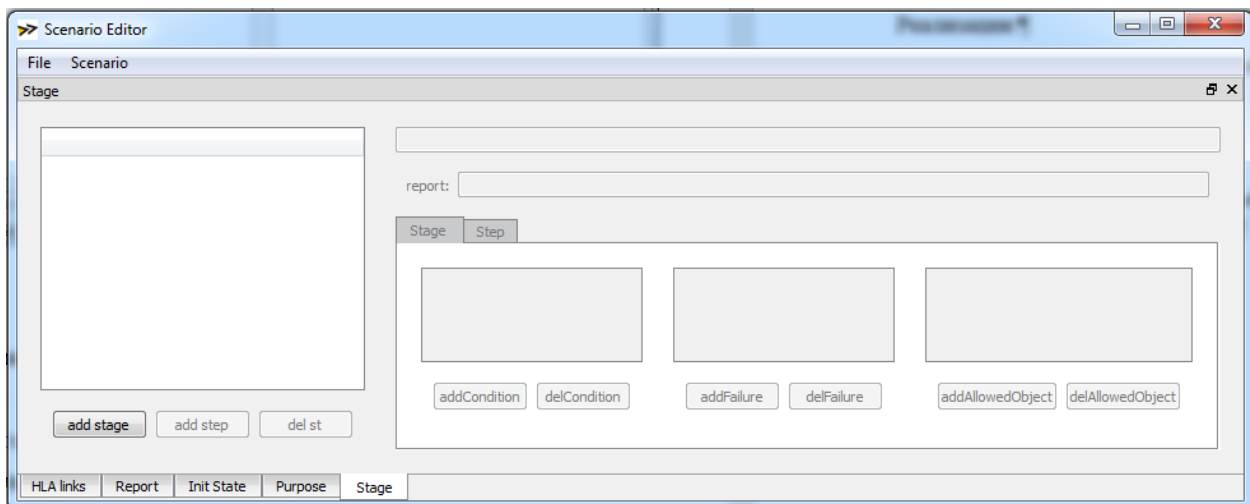


Рис. 13 Редактор сценариев вкладка Stage

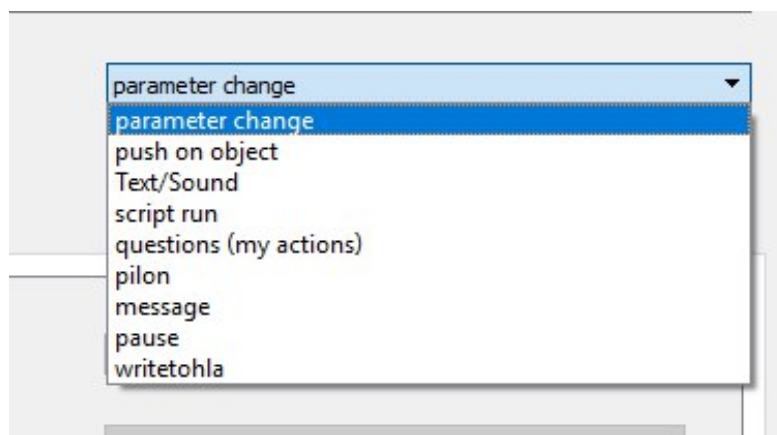


Рис. 14 Редактор сценариев типы шагов

Типы шагов:

1. Parameter change

Условие прохождения шага, связанное с изменениями физических величин в указанных аппаратах.

2. Push on object

Целью шага является нажатие на объект в 3D модели.

3. Text/sound

Вывод текста на экран/проигрывание звука.

4. Script run

Запуск скрипта, по имени который выполняет действие в 3D модели.

5. Question

Вывод вопроса на экран с 4 вариантами ответа.

6. Pilon

Ставит в 3D пилон который указывает позицию куда нужно переместиться.

7. Pause

Останавливает выполнение сценария на заданный промежуток времени.

8. Write HLA

Выставляет указанные параметры у объекта на сервере.

Реализация.

Программа для редактирования и создания сценариев для обучающего имитационного тренажёра реализована на языке C++ [2] с использованием библиотеки Qt 5.1.1.

Демонстрация элементов программного кода.

Создание виджетов на главной форме.

```
{
    dockPurpose = new QDockWidget(tr("Purpose"), this);
    dockPurpose->setObjectName("PurposeDock");
    dockPurpose->setAllowedAreas(Qt::AllDockWidgetAreas);

    Purpose = new PurposeWidget(varGroups, purposes);
    Purpose->setObjectName("Purpose");
    dockPurpose->setWidget(Purpose);

    addDockWidget(Qt::LeftDockWidgetArea, dockPurpose);
}

//HLA
{
    dockHLATable = new QDockWidget(tr("HLA links"), this);
    dockHLATable->setObjectName("HLALinks");
    dockHLATable->setAllowedAreas(Qt::AllDockWidgetAreas);
    HLATable = new HLATableWidget(varGroups);
    HLATable->setObjectName("HLATable");
    dockHLATable->setWidget(HLATable);

    connect(this, SIGNAL(StartHLA_run(QString)), HLATable,
SLOT(StartHLA_run(QString)));
    connect(HLATable, SIGNAL(newGroup_signal(TreeGroup)), this,
SLOT(newGroup_slot(TreeGroup)));
    connect(HLATable, SIGNAL(newVar_signal(QString, TreeVariable)),
this, SLOT(newVar_slot(QString, TreeVariable)));
    connect(HLATable, SIGNAL(hla_stop()), this, SLOT(hla_stop()));

    addDockWidget(Qt::RightDockWidgetArea, dockHLATable);
}

//InitState
{
    dockInitstate = new QDockWidget(tr("Init State"), this);
    dockInitstate->setObjectName("InitState");
    dockInitstate->setAllowedAreas(Qt::AllDockWidgetAreas);

    initstate = new InitState(dataInitState);
    Purpose->setObjectName("InitState");
    dockInitstate->setWidget(initstate);

    addDockWidget(Qt::LeftDockWidgetArea, dockInitstate);
}
```

```

}

//Этапы и шаги
{
    dockStage = new QDockWidget(tr("Stage"), this);
    dockStage->setObjectName("StageDock");
    dockStage->setAllowedAreas(Qt::AllDockWidgetAreas);

    Stage = new StageWidget(stagestep, varGroups);
    Stage->setObjectName("Stage");
    dockStage->setWidget(Stage);

    addDockWidget(Qt::LeftDockWidgetArea, dockStage);
}

Проверка выполнения шагов сценария.
#pragma region Выполнение этапов/шагов
for (int st = 0; st < stagestep->stages.size(); st++)
{
    if (stagestep->stages.at(st).complete == true) continue;

    if (stagestep->stages.at(st).sent == false)
    {
        if (stagestep->stages.at(st).report != "")
        {
            Report->addRecord(tr("Start stage: ") + stagestep-
>stages.at(st).report, "");
        }
        stagestep->stages.at(st).sent = true;
    }

    bool stage_condition = true;
    bool stage_failure = true;

    //цели этапа
    for (int i = 0; i < stagestep->stages.at(st).conditions.size(); i++)
    {
        for (int g = 0; g < varGroups->Variables.size(); g++)
        {
            if (varGroups->Variables.at(g).discription != stagestep-
>stages.at(st).conditions.at(i).group) continue;

            for (int v = 0; v < varGroups-
>Variables.at(g).variables.size(); v++)
            {
                if (varGroups-
>Variables.at(g).variables.at(v).discription != stagestep-
>stages.at(st).conditions.at(i).variable) continue;

                //bool
                if (stagestep->stages.at(st).conditions.at(i).type
== ConditionAndFailureClass::boolean)
                {
                    bool hla_v = false;
                    if (varGroups-
>Variables.at(g).variables.at(v).hla_value > 0) hla_v = true;

                    if (stagestep-
>stages.at(st).conditions.at(i).boolean_value == hla_v)
                    {
                        //complete
                        stagestep-
>stages.at(st).conditions.at(i).complete = true;

```

```

        }
        else
        {
            //not complete
            stagestep-
>stages.at(st).conditions.at(i).complete = false;
        }
    }
    //number
    if (stagestep->stages.at(st).conditions.at(i).type
== ConditionAndFailureClass::number)
    {
        double        hla_v        =        varGroups-
>Variables.at(g).variables.at(v).hla_value;

        if (stagestep-
>stages.at(st).conditions.at(i).number_type
ConditionAndFailureClass::number_left)
        {
            if (stagestep-
>stages.at(st).conditions.at(i).leftNumber_value <= hla_v)
            {
                stagestep-
>stages.at(st).conditions.at(i).complete = true;
            }
            else
            {
                stagestep-
>stages.at(st).conditions.at(i).complete = false;
            }
        }
        if (stagestep-
>stages.at(st).conditions.at(i).number_type
ConditionAndFailureClass::number_right)
        {
            if (stagestep-
>stages.at(st).conditions.at(i).rightNumber_value >= hla_v)
            {
                stagestep-
>stages.at(st).conditions.at(i).complete = true;
            }
            else
            {
                stagestep-
>stages.at(st).conditions.at(i).complete = false;
            }
        }
        if (stagestep-
>stages.at(st).conditions.at(i).number_type
ConditionAndFailureClass::number_both)
        {
            if (stagestep-
>stages.at(st).conditions.at(i).leftNumber_value <= hla_v && stagestep-
>stages.at(st).conditions.at(i).rightNumber_value >= hla_v)
            {
                stagestep-
>stages.at(st).conditions.at(i).complete = true;
            }
            else
            {
                stagestep-
>stages.at(st).conditions.at(i).complete = false;
            }
        }
    }
}

```

```

        }
    }

    if (stagestep->stages.at(st).conditions.at(i).complete == false) stage_condition = false;

    QList<QListWidgetItem *> items = Stage->listCondition->findItems(stagestep->stages.at(st).conditions.at(i).discription,
Qt::MatchExactly);

    if (items.size() > 0) {
        if (stagestep->stages.at(st).conditions.at(i).complete == true)
        {
            items.at(0)->setForeground(Qt::darkGreen);
        }
        else
        {
            items.at(0)->setForeground(Qt::red);
        }
    }

    break;
}
break;
}
}

```

Заключение

Имея реализованный имитационный компьютерный тренажёр, который будет в точности повторять аутентичную площадку, открываются большие возможности для обучения персонала порядку действий по технологическому регламенту, действиям при возникновении аварийных ситуаций на объектах и получения навыков работы на имитируемом оборудовании. Также имитационный компьютерный тренажёр решает проблему доступности промышленных объектов для обучаемых.

СПИСОК ЛИТЕРАТУРЫ

1. Гаммер М.Д. Сызранцев В.Н. Имитаторы на базе программно-аппаратной платформы в техническом образовании. - Тюмень: ТюмГНГУ, 2010. - 271 с.
2. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений на С++ / Г. Буч; Пер. с англ. - 2-е изд. - М.: Изд-во «Бином», СПб.: «Невский диалект», 2000.- 560 с.

3. Страуструп Б. Язык программирования C++. Спец.изд./ Б. Страуструп; пер. с англ. - М.: Изд-во Бином, СПб.: Невский диалект, 2000. – 1099 с.
4. Современные звуковые технологии в играх [Электронный ресурс]. - Электрон. дан. - Режим доступа: <http://www.ixbt.com:80/multimedia/sound-technology-in-games-2003>
5. Хилл Ф. OpenGL. Программирование компьютерной графики / Ф. Хилл. - СПб.: Питер, 2002.- 1088с.
6. Эйнджел Эдвард. Интерактивная компьютерная графика. Вводный курс на базе OpenGL, 2 изд.: Пер. с англ / Эдвард Эйнджел. - М.: Издательский дом “Вильямс”, 2001. - 592с.: ил.
7. Язык программирования Lua [Электронный ресурс]. - Электрон. дан. - Режим доступа: <http://ru.wikipedia.org/wiki/Lua>