

В. И. МОЙСЕЕНКО, В. В. РАДЧЕНКО (УкрГАЗТ)

МОДЕЛИРОВАНИЯ ПОЕЗДНОЙ СИТУАЦИИ ДЛЯ ИНФОРМАЦИОННО УПРАВЛЯЮЩИХ СИСТЕМ ЖЕЛЕЗНОДОРОЖНОГО ТРАНСПОРТА

Розглянуто питання моделювання поїзної ситуації на станції. Була побудована математична модель роботи пристроїв залізничної автоматики та поїзних пересувань. Описано спеціальне програмне забезпечення, яке дозволяє спростити та прискорити процес налагодження систем у різноманітних режимах.

Рассмотрены вопросы моделирования поездной ситуации на станции. Была создана математическая модель работы устройств железнодорожной автоматики и поездных передвижений. Описано специальное программное обеспечение, позволившее упростить и ускорить процесс отладки систем в различных режимах.

The article considers the issues of modelling train situations at stations and develops a mathematical model of operation of railway automatics devices and movements of trains. The special software has been described, which allows essentially to simplify and speed up the process of systems arrangement in different modes.

Стремительное развитие вычислительной техники позволило создавать программно-аппаратные комплексы способные реализовывать любые технологические процессы. На данном этапе железная дорога остается одним из самых привлекательных объектов автоматизации. Технологические процессы довольно сложны и человек уже не справляется с таким большим количеством информации. Кроме того, установленные системы устарели и требуют замены. Дальнейшее развитие функций этих систем не возможно, поскольку элементная база устарела [1; 2].

Новым витком в развитии железнодорожной автоматики стало активное внедрение передовых электронно-вычислительных систем, позволяющих реализовать технологические процессы любой сложности.

Как правило, технологический процесс работы станции довольно сложен и имеет большое количество составляющих. По этой причине разработка систем становится очень трудоемкой. Приходится учитывать большое количество нюансов связанных с особенностью функционирования реального объекта.

Известны работы ряда авторов [9], в которых моделирование технологического процесса ведется под действующие объекты, такой подход практически полностью исключает возможность отладки разрабатываемой системы на объекте, что приводит к существенному увеличению продолжительности пуско-наладочных работ. Это связано с тем, что постоянно по-

ступающий поток информации мешает проследить реакцию системы. В отдельных случаях имеют место ситуации, при которых возникает нехватка, либо полное отсутствие необходимой информации.

Целью статьи является создание унифицированной модели объекта управления и контроля, которая бы обеспечивала возможность отладки. Сложность обычного моделирования определяется многозадачностью технологического процесса, который реализует система управления движением поездов.

При создании единого комплекса технических средств станции Новая Бавария разработчики столкнулись с проблемой отладки программного обеспечения (ПО) системы. Комплекс, включал в себя систему управления движением поездов, систему автоматического оповещения и автоматизированное рабочее место электромеханика.

Некоторая информация была необходима для всех трех систем, некоторая обрабатывалась только одной из них. К примеру, контроль о состоянии путевого развития необходим всем системам, а информация о напряжениях на питающем и релейном концах рельсовых цепей и лучевых реле необходима только для рабочего места электромеханика. Управляющие воздействия формировались только рабочим местом дежурного по станции.

Нижний уровень системы образован контроллерами, получающими информацию от объектов, и главными контроллерами, играющими роль серверов системы. Верхний уровень

образовывают автоматизированные рабочие места персонала станции. Разработка и отладка систем шла параллельно. В случае ошибочной работы системы приходилось выяснять достоверность входных данных, что затрудняло процесс поиска ошибок.

Для реализации программных задач верхнего уровня создавалось специальное программное обеспечение, которое позволило моделировать информацию, поступающую от главных контроллеров, и реагировать на команды управления, передаваемые нижнему уровню системы [3]. Такой подход позволил отделить процесс отладки верхнего уровня от подсистем нижнего уровня.

Разработка модели станции разбивалась на три этапа. На первом этапе происходило обучение модели передаче основных информационных посылок.

В специальном файле настроек были описаны коды команд для каждого объекта контроля. Это сделало модель довольно гибкой, так как добавление новых информационных сообщений сводилось к текстовому описанию команды и ее шестнадцатиричного кода. Ниже приведен фрагмент команд, описывающих работу рельсовой цепи

```
<TCCommand value="0000000F" name="СП
под ток" shortname="SP_UP" />
<TCCommand value="FFFFFFF0" name="СП
без тока" shortname="SP_Down" />
<TCCommand value="000000F0" name="З
без тока" shortname="Z_UP" />
<TCCommand value="FFFFFFF0F" name="З
под ток" shortname="Z_Down" />
<TCCommand value="000000FF" name="РИ
под ток" shortname="RI_UP" />
<TCCommand value="FFFFFFF00" name="РИ
без тока" shortname="RI_Down" />
```

Кроме основных сообщений для всех объектов определенного типа была возможность добавления персональных команд. В качестве примера ниже представлено описание команды для рельсовой цепи НПП.

```
<TrackCircuit name="НПП" id="1099">
<Param begin="17" objs="17" type="real"
side="odd" />
<Command value="0000FFFF" name="NNP
Command" />
</TrackCircuit>
```

Формирование списка основных команд осуществлялось на основании синтезированных графов переходов. Для примера на рис. 1 представлен граф переходов рельсовой цепи.

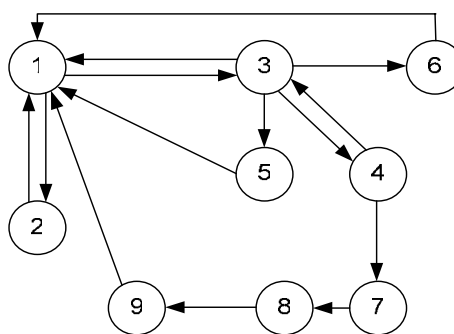


Рис. 1. Граф переходов рельсовой цепи:

- 1 – рельсовая цепь свободна, не замкнута, не размыкается;
- 2 – рельсовая цепь занята, не замкнута, не размыкается;
- 3 – рельсовая цепь свободна и замкнута в маршруте;
- 4 – занятие рельсовой цепи замкнутой в маршруте;
- 5 – отмена маршрута; 6 – искусственное размыкание при свободной рельсовой цепи; 7 – искусственное размыкание при занятой рельсовой цепи; 8 – отказ в работе рельсовой цепи; 9 – ремонт

В процессе анализа сформированы все возможные переходы из одного состояния в другое и выявлены все причины этих переходов. В последствии графы реализовывались в программном обеспечении модели [4–6]. Для примера ниже представлен код реализации данного графа на языке C++ [7; 8]

```
void SwitchSect::StateAdapt()
{
    ushort state;
    state = NO_DEFINED;
    if( _Coils[0] == SP_UP )
        state = SECT_FREE;
    else if( _Coils[0] == SP_DOWN )
        state = SECT_BUSY;
    if( _Coils[1] == Z_DOWN )
    {
        if( state == SECT_FREE )
            state = SECT_ROUTE;
        else if( state == SECT_BUSY )
            state = SECT_BROUTE;
    }
    if( _Coils[2] == RI_UP )
    {
        if( state == SECT_FREE ||
state == SECT_ROUTE )
            state = SECT_UCFREE;
        else if( state == SECT_BUSY || state
== SECT_BROUTE )
            state = SECT_UCBUSY;
    }
    SetState(state);
}
}
```

Данное ПО включало в себя сервер системы и позволяло подключать все рабочие места одновременно (рис. 2). Подключившееся рабочее место получало информацию о состоянии путе-

вого развития на момент подключения. Сервер позволял не только передавать сообщения рабочим местам, но и обмен сообщениями. Одно рабочее место могло передавать информацию

другому. К примеру, автоматическая система оповещения передавала рабочему месту дежурного по станции информацию об установке и отмене фронта работ и т. д.

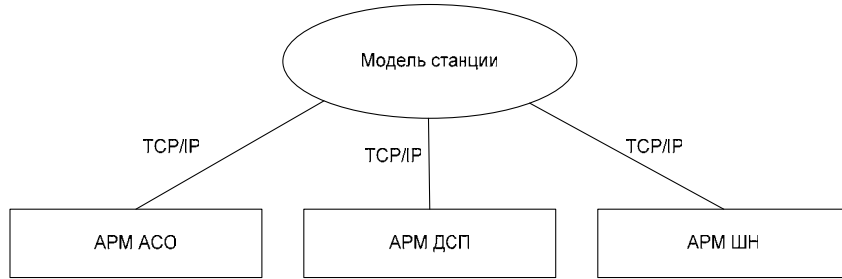


Рис. 2. Структура подключений АРМ к модели станции

Для передачи команды в основном окне выбирался объект и указывалось необходимое сообщение. При этом предусматривалась возможность указания задержки передачи выбранной команды. За счет этого обеспечивается передача необходимых последовательностей сообщений, состоящих из нескольких команд, посылаемых через определенные промежутки времени.

системе организовать обмен сообщениями без потери очередности, а также производить повторы посылок до их получения. В модели квитанции использовались для отображения информации на станции. Такой принцип обеспечивал наглядность получения команд при передаче в процессе отладки.

Передача каждой команды от АРМ серверу системы и в обратном направлении завершалась квитированием (рис. 3). Это позволяло

Основное окно модели представляет собой поле, содержащее путевое развитие станции и световую индикацию табло (рис. 4). Любой из элементов путевого развития можно активизировать нажатием кнопки манипулятора «мышь».

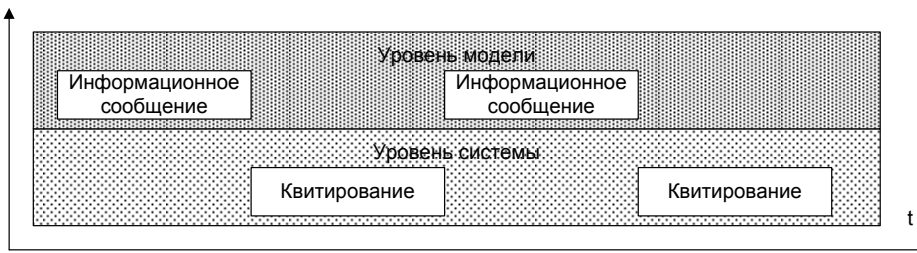


Рис. 3. Диаграмма передачи информационных сообщений

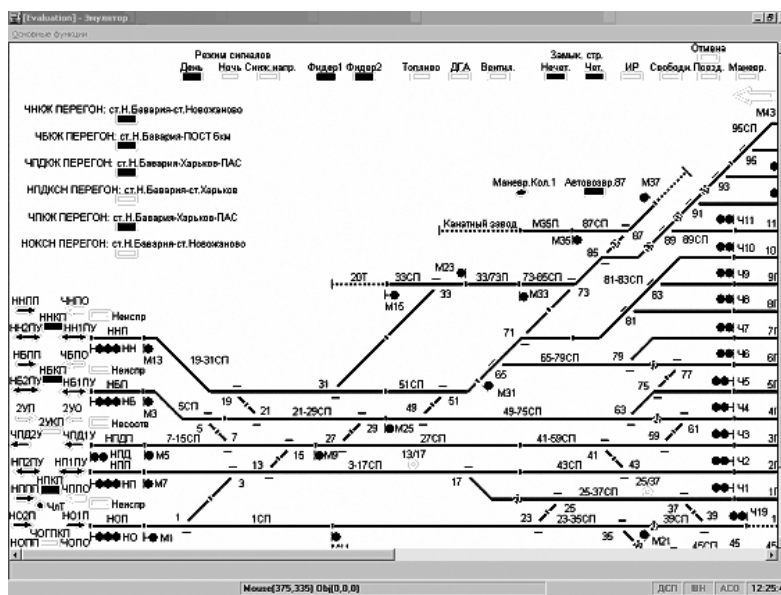


Рис. 4. Основной вид приложения

Передача основных информационных сообщений позволяет отладить функции по обработке входящей информации и ее представлении. Подавая необходимые последовательности, в нужный момент можно проследить реакцию системы. При этом вопрос о достоверности информации уже не стоит. Оператор сам решает, какие именно команды надо передать системе. Облегчается процесс внесения неисправности в

работу системы путем формирования заведомо ложного сообщения. На рис. 5 представлен фрагмент видеogramмы станции с установленным маршрутом отправления.

На рис. 6 изображено окно с командами по одному из объектов. У каждого элемента путевого развития свой перечень команд. Данный режим передачи сообщений назван ручным или пошаговым.

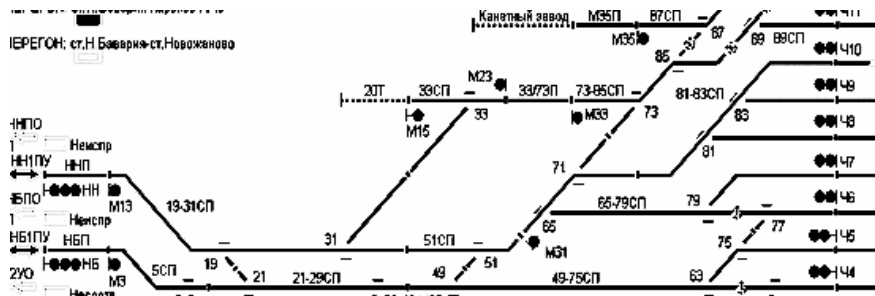


Рис. 5. Маршрут отправления

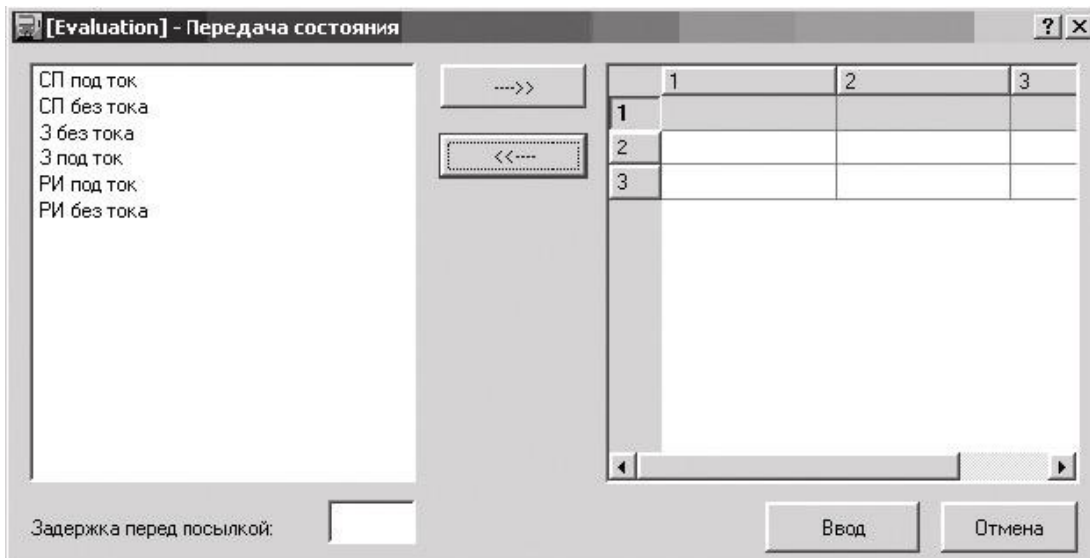


Рис. 6. Список команд для рельсовой цепи

На втором этапе ставилась задача по обучению модели реагировать на основные управляющие команды. При установке маршрута приходилось производить очень много манипуляций. В связи с этим производилось обучение модели самостоятельно реагировать на управляющие воздействия. Так, к недостатку ручного режима относится достаточно большое число операций, выполняемых в процессе установки маршрута.

При получении команды на перевод стрелки устанавливалось под ток соответствующее управляющее реле. После чего обесточивалось одно контрольное реле, находящееся до этого под током, и затем формировалась команда на возбуждение другого контрольного реле.

После этого были сформированы реакции модели на формирование сложных команд, таких как установка маршрутов. При этом передавались информационные сообщения о переводе стрелок, замыкании секций и открытии светофора. Это существенно сократило время на установку маршрута. Данный режим можно отнести к автоматическому.

Модель позволила формировать сообщения с произвольными временными интервалами между послылками. Что позволило проверить работу подпрограмм, связанных с измерением временных характеристик компонентов системы. К примеру, время перевода стрелки, измерение замедления сигнального реле светофора или выдержку времени при отмене маршрута.

После отладки основных функций систем при помощи модели можно было подключать их к реально действующему объекту.

Испытания на реальном объекте выявили, что не все части программ были отлажены. В определенных ситуациях система выдавала ложную информацию. Отладить работу можно было, только передавая команды, приводящие к сбоям.

Каждая из систем вела свой «черный ящик». Вся поступающая информация записывалась в

отдельный файл на диске. На основе этого создан третий режим – режим «черного ящика». При обнаружении ошибочной работы системы достаточно было запомнить время, когда система неправильно отреагировала. После этого «черный ящик» открывался моделью, и все команды последовательно передавались системе. Таким образом, можно было определить команду и ситуацию, когда система начинала работать некорректно. Окно «черного ящика» представлено на рис. 7.

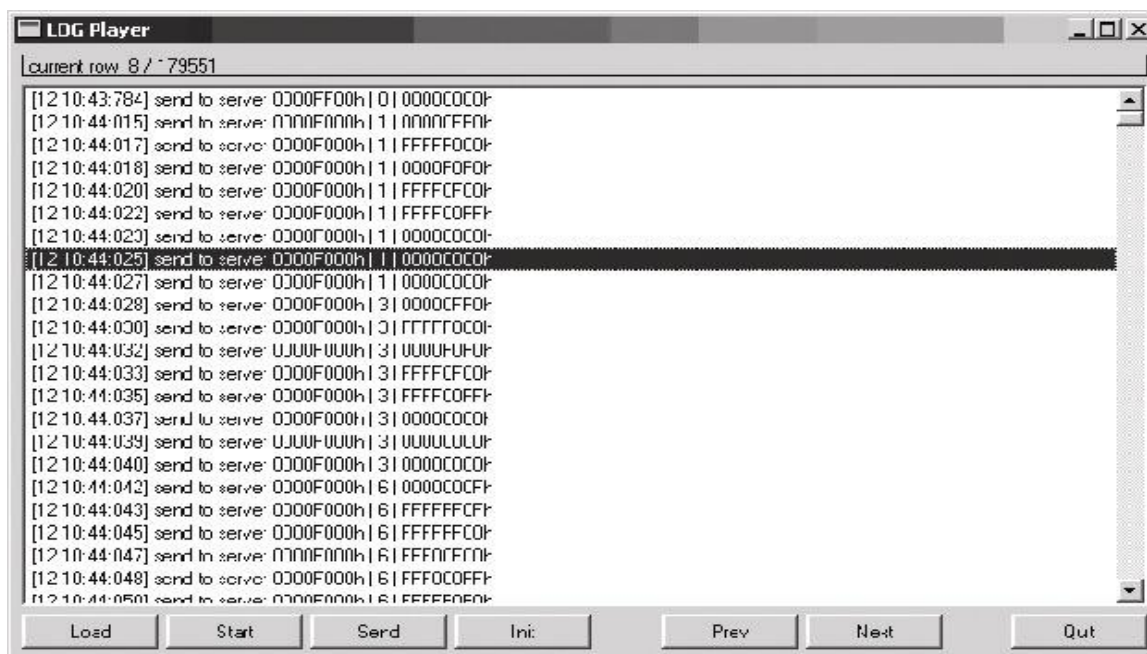


Рис. 7. Проигрыватель черного ящика

Созданная модель позволила облегчить процесс отладки программных и аппаратных средств и существенно сократить время на разработку системы в целом.

Авторами разработана модель, описывающая основные технологические процессы крупных железнодорожных станций. На основе синтезированных графов были получены модели поведения объектов управления и контроля.

Предложенный подход позволяет существенно сократить период отладки и ввода в эксплуатацию автоматизированных систем управления движением поездов. Это позволило выполнить пуско-наладочные работы, не нарушая технологического процесса работы станции.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Глушков В. М. Синтез цифровых автоматов. – М.: Физматгиз, 1962
2. Сапожников В. В. Дискретные устройства железнодорожной автоматики, телемеханики и

- связи / В. В. Сапожников, Ю. А. Кравцов, В. В. Сапожников. – М.: Транспорт, 1988.
3. Трухаев Р. И. Модели принятия решений в условиях неопределенности. – М.: Наука, 1981. – 258 с.
4. Амосов Н. М. Нейрокомпьютеры и интеллектуальные роботы / Н. М. Амосов, Т. Н. Байдык, А. Д. Гольцев. – К.: Наук. дум., 1994 – 272 с.
5. Борисов А. Н. Обработка нечеткой информации в системах принятия решений / А. Н. Борисов, А. В. Алексеев, Г. В. Меркурьева и др.
6. Hecht – Nielsen R/ Neurocomputing: picking the human brain. // IEEE SPECTRUM 1998 – V.25.N3 – p. 36–41.
7. Krten, Rob, 1965 – Getting started with QNX Neutrino 2 – a guide for realtime programmers.
8. Bjarne Stroustrup AT&T Labs Murray Hill, New Jersey, The C++ programming language.
9. Чепцов М. Н. Динамическая поездная модель района диспетчерского управления. Дис... канд. техн. наук. – Харьков, 2001.

Поступила в редколлегию 12.10.2005.