электронный журнал
МОЛОДЕЖНЫЙ НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК
Издатель ФГБОУ ВПО "МГТУ им. Н.Э. Баумана". Эл No. ФС77-51038.

 # 11, ноябрь 2015

УДК 004.825

Системы, основанные на знаниях (на примере системы защиты от несанкционированного доступа)

***Чернецкая Д.И.**, магистр*
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы автоматического управления»

Научный руководитель: Суханов В.А., к.т.н, доцент
Россия, 105005, г. Москва, МГТУ им. Н.Э. Баумана,
кафедра «Системы автоматического управления»
suhanov@iu1.bmstu.ru

В настоящее время интеллектуальные системы являются неотъемлемыми компонентами развитых автоматизированных систем. Среди основных моделей представления знаний в интеллектуальных системах используются системы продукций, системы фреймов, семантические сети, ленемы, нейронные сети. Продукционные модели можно считать наиболее распространенными моделями представления знаний. Для представления набора продукционных правил и структуризации интеллектуальной системы могут использоваться сети Петри.

Целью данной работы является разработка продукционной модели представления знаний на примере конкретной предметной области – системы защиты от несанкционированного доступа.

Задачи исследования, вытекающие из поставленной цели:

1. изучить подходы к различным видам СОЗ;
2. представить знания на основе продукционной модели;
3. реализовать алгоритм на основе рекуррентного уравнения, описывающего динамику процесса управления, с использованием математического аппарата сетей Петри;
4. разработать программное приложение для моделирования и апробации предложенной системы;

Структура системы, основанной на знаниях (СОЗ) существенно зависит от ее назначения (цели), которое можно определить следующей совокупностью параметров:

1. Цель создания СОЗ – обучение специалистов, решение задач, автоматизация рутинных работ, тиражирование знаний экспертов и т.д.;
2. Основной пользователь – неспециалист в проблемной области, специалист, учащийся и т.п.;
3. Тип проблемной области – статическая, динамическая и т.д.

Общая структура СОЗ содержит несколько функциональных блоков (подсистем), часть которых объединена в функциональные группы (интеллектуальные модули, пользовательский интерфейс). В структуре СОЗ можно выделить три основных укрупненных блока: базу данных, базу знаний и решающий блок.

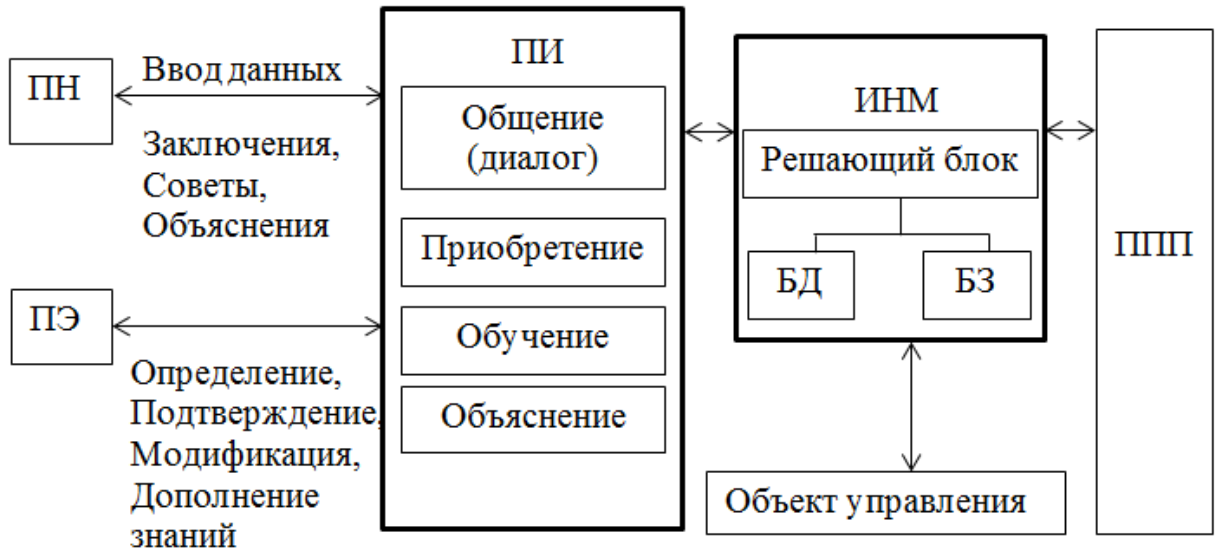

Рис. 1. Обобщенная функциональная схема СОЗ

На рисунке 1 представлено: ПН – пользователь-неспециалист; ПЭ – пользователь-эксперт, ПИФ – пользовательский интерфейс; ИНМ – интеллектуальные модули, общение (диалог) – диалоговый компонент; обучение – компонент обучения; приобретение – компонент приобретения знаний; объяснение – компонент объяснения; ППП – пакеты прикладных программ.

В настоящее время разработано множество моделей представления знаний, но наиболее распространенными являются продукционные модели. Продукционная модель – это модель, основанная на правилах, позволяющая представить знание в виде предложений типа: «ЕСЛИ условие, ТО действие».

Любое правило состоит из одной или нескольких пар «атрибут-значение». В рабочей памяти систем, основанных на продукционных моделях, хранятся пары атрибут-

значение, истинность которых установлена в процессе решения конкретной задачи к некоторому текущему моменту времени. Содержимое рабочей памяти изменяется в процессе решения задачи. Это происходит по мере срабатывания правил. Правило срабатывает, если при сопоставлении фактов, содержащихся в рабочей памяти, с антецедентом анализируемого правила имеет место совпадение, при этом заключение сработавшего правила заносится в рабочую память. Поэтому в процессе логического вывода объём фактов в рабочей памяти, как правило, увеличивается (уменьшаться он может в том случае, если действие какого-нибудь правила состоит в удалении фактов из рабочей памяти). В процессе логического вывода каждое правило из базы правил может сработать только один раз.

Существуют два типа продукционных систем – с «прямыми» и «обратными» выводами. Прямые выводы реализуют стратегию «от фактов к заключениям». При обратных выводах выдвигаются гипотезы вероятностных заключений, которые могут быть подтверждены или опровергнуты на основании фактов, поступающих в рабочую память. Существуют также системы с двунаправленными выводами.

Основные достоинства систем, основанных на продукционных моделях, связаны с простотой представления знаний и организации логического вывода. К недостаткам таких систем можно отнести следующее:

1. Неясность взаимных отношений правил;
2. Сложность оценки целостного образа знаний;
3. Низкая эффективность обработки знаний.

При разработке небольших систем (десятки правил) проявляются в основном положительные стороны продукционных моделей знаний, однако при увеличении объёма знаний более заметными становятся слабые стороны.

Сети Петри (СП) – это математический аппарат двудольных графов, используемый для анализа, моделирования и представления причинно-следственных связей в сложных распределенных дискретных системах, процессах параллельной обработки и синхронизации, что является весьма важным, в частности, для продукционных систем. При этом общее описание процессов может быть представлено в форме двудольных ориентированных графов. Графы переходов и конечные автоматы являются частными случаями СП.

Важно, что СП описывают не только сам процесс, но и управление этим процессом.

Представление набора продукционных правил и структуризация их в форме СП могут использоваться в качестве способа внутреннего представления знаний в СОЗ.

СП представляет собой граф, который обладает следующими свойствами:

1. Каждая из вершин графа является либо позицией, либо переходом;
2. Граф является ориентированным, каждая дуга соединяет позицию с переходом, либо переход с позицией;
3. Каждой из позиций ставится в соответствие некоторое целое неотрицательное число N, при этом говорят, что в этой позиции находится N меток; это число может быть равно 0, т. е. меток в позиции нет; в процессе работы сети, количество меток в позиции может изменяться;
4. Каждой дуге ставится в соответствие некоторое натуральное число, которое называют кратностью данной дуги; кратность 1 обычно опускается; иногда для общности вводят кратность 0, которая соответствует отсутствию дуги; кратность дуги в процессе работы сети не изменяется.

Формально СП может быть представлена в виде четверки G = (P, T, I, O) или графически в форме двудольных ориентированных графов (на рисунке 3.), где P ={p1, ..., pn}, P≠0 – множество вершин, называемых позициями (местами);T ={t1,..., tm}, T≠0 – множество вершин, называемых переходами; I, O – входная и выходная функции, которые представляют собой соответственно прямую и обратную функции (матрицы) инцидентности:

I: PxT→ {0, 1} – описывает входные для переходов дуги;

O: TxP→ {0, 1} - описывает выходные для переходов дуги.

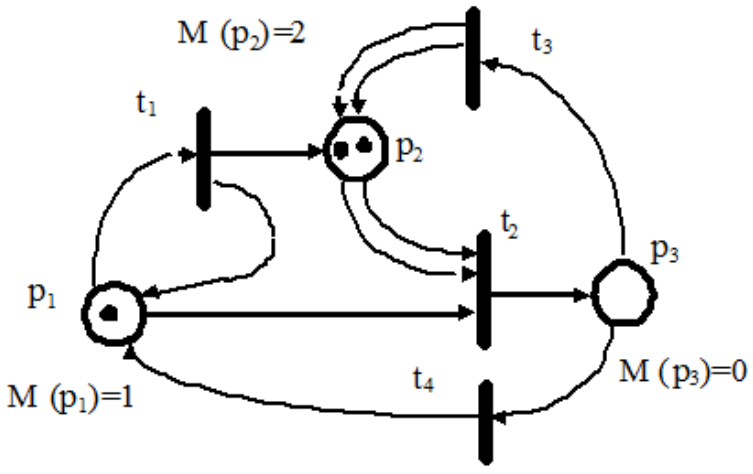

Рис. 2. Графическое представление СП

Таким образом, отношение инцидентности можно задать, указав для каждого перехода t_i два вектора – $I(t_i)$ и $O(t_i)$ размерностью n, где $I(t_i)$ содержит сведения о входных для t_i дугах для всех позиций сети, $O(t_i)$ – о выходных дугах.

Для каждого перехода t_j можно определить множества входных $I(t_j)$ и выходных $O(t_j)$ позиций:

$I(t_j)=\{p_i \in P/I(p_i, t_j)=1\}$, $O(t_j)=\{ p_i \in P/O(t_j,p_i)=1\}$, где $i=\overline{1,n}$, $j=\overline{1,m}$.

Если дуга двудольного графа направлена из i-ой позиции в j-ый переход, то $I(p_i,t_j)=1$, иначе $I(p_i, t_j)=0$, где $i=\overline{1,n}$, $j=\overline{1,m}$. Аналогично – для других элементов.

Для описания динамических свойств сети (моделируемой системы) вводится функция M разметки сети M: $P \rightarrow \{0,1,2,\ldots\}$. С помощью функции $M_i= M(p_i)$, где $i=\overline{1,n}$, позиции помечаются целыми неотрицательными числами. При графическом задании СП разметка отображается соответствующим числом точек $M_i=M(p_i)$ внутри соответствующих кружков – позиций p_i. Эти точки называют метками (фишками, маркерами).

Разметку можно представить также в векторной форме:

$M = (M_1, M_2,..., M_n)$,

где $M_i=M(p_i)$.

Размеченная сеть обозначается в виде:

$G=(P,T,I,O,M_0)$,

где M_0 - начальная разметка.

Срабатывание какого-либо перехода t_j в размеченной сети ведет к смене разметки.

Аналогично для каждой позиции p_i можно определить множества входных и выходных переходов –$I(p_i)$ и $O(p_i)$ соответственно, $i=\overline{1,n}$.

Можно получить математическое описание функционирования СП в виде уравнения, описывающего изменение разметки в результате срабатывания перехода. Новая разметка M(k+1), т.е. разметка после срабатывания перехода t_j, может быть определена по следующему правилу:

$M(k+1) = M(k) - I(t_j) + O(t_j) = M(k) + \Phi(t_j)$,

где M(k) – вектор текущей разметки (состояние СП); $\Phi(t_j) = O(t_j) - I(t_j)$ – столбец матрицы Ф, соответствующий переходу t_j.

Это уравнение описывает изменение разметки сети при срабатывании одного j-го перехода, т.е. не носит общего характера.

Срабатывание перехода –неделимое действие, изменяющее разметку входных и выходных позиций, при этом событие, изображаемое переходом, изменяет состояние

(емкость) непосредственно связанных с ним условий так, что емкость предусловий, вызвавшихреализацию этого события, уменьшается, а емкость постусловий, на которые оно влияет, увеличивается. Функционирование размеченной СП представляет собой поочередный запуск переходов. В каждый момент времени какие-то переходы могут быть запущены, а какие-то – нет. Другими словами, это процесс изменения разметки (начиная от M_0), осуществляемый по правилам:

1. Если для перехода t_j выполняется необходимое условие срабатывания, то происходит срабатывание этого перехода, в результате чего изменяется разметка во всех входных и выходных позициях только этого перехода:
    a. Из каждой позиции, которая является входной для данного перехода, изымается число меток, равное кратности соответствующей дуги;
    b. В каждую выходную для данного перехода позицию добавляется число меток, равное кратности соответствующей дуги;
2. Если два или более перехода могут одновременно сработать, и они не имеют общих входных позиций, то их срабатывание независимо и осуществляется в любой последовательности или параллельно;
3. Если несколько переходов могут сработать и имеют общую входную позицию, то сначала срабатывает только один переход, любой из них; при этом может оказаться, что, сработав, этот переход лишит другие переходы возможности сработать - этим моделируется конфликтная ситуация (она устраняется вне формализма СП);
4. Функционирование СП продолжается до тех пор, пока существует хотя бы один переход, готовый к срабатыванию, и останавливается, если ни один из ее переходов не может сработать.

Система защиты от несанкционированного доступа (ЗНД) находится в начальном состоянии, когда соответствующие механизмы, обеспечивающие предотвращение несанкционированного доступа, приведены в исходное состояние, т.е. доступ в охранную зону недоступен.

Для снятия защиты, чтобы войти в зону, необходимо выполнить следующие действия:

1. Ввести пароль
2. Повернуть ручку двери.
3. Если код правильный, то замок должен открыться, войти в дверь и дождаться закрытия двери, чтобы обеспечить возврат системы в исходное состояние;

4. При попытке открывания двери с неправильно набранным кодом пароля система должна включить тревожную сигнализацию (сирена и лампа);
5. Возможен вариант открытия двери без ввода пароля.

В системе ЗНД можно выделить несколько устойчивых и повторяющихся состояний, что позволяет составить общую блок-схему ее функционирования:

1. Начало (исходное/начальное состояние);
2. Ввод и сравнение кодов;
3. Ожидание открытия двери;
4. Ожидание закрытия двери;
5. Тревожная сигнализация.

В исходном состоянии система может находиться неопределенное время, пока не будет начат ввод пароля, когда система осуществляет переход в состояние ввода и сравнения кодов.

Продукционный подход к представлению знаний и их структурированию на основе аппарата СП позволяют сформировать дискретную динамическую модель системы ЗНД, которая одновременно является и схемой/механизмом управления этой системой. Для этого необходимо сформировать базы данных и знаний, а также систему управления ими.

Вектор состояния модели представляет собой вектор разметок всех позиций в СП, поэтому его размерность равна числу позиций в СП.

База данных представлена в виде массива натуральных чисел (разметки отдельных позиций), в котором хранится множество номеров помеченных (входных и выходных) позиций С. На основании этих знаний и содержимого матриц I и O можно определить переходы/продукции, которые могут сработать. После этого организовывается выбор одного из них.

Зная текущую разметку и вектор управления, с помощью рекуррентного уравнения можно определить следующий вектор разметки (вектор состояния объекта).

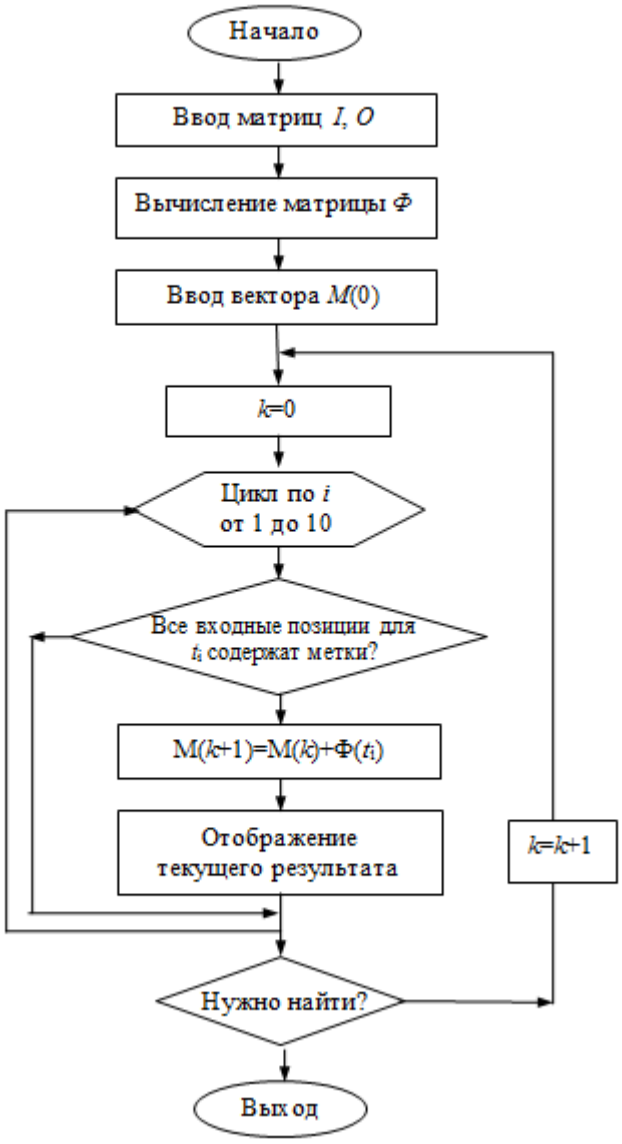

Рис. 3. Блок-схема функционирования дискретного управляющего устройства

СП, реализующая возможные сценарии в системе ЗНД, приведена на рис. 4. Для данной СП получены уравнения, описывающие динамику процесса управления в форме рекуррентного матричного выражения, которое положено в основу моделирования и реализации конкретных сценариев.

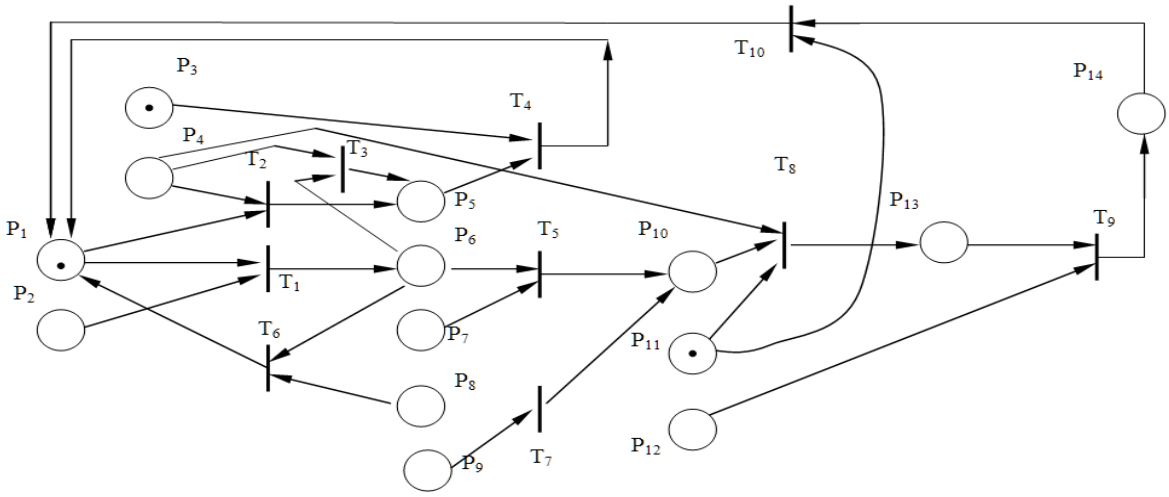

Рис. 4. Реализация всех возможных сценариев в системе ЗНД на основе СП

Таблица 1

Описание позиций СП

Номер позиции	Интерпретация
1	Исходное/начальное состояние системы (отключена сирена, отключена лампа)
2	Начат ввод пароля
3	Ручка двери не нажата
4	Ручка двери нажата
5	Включить сирену и лампу
6	Сравнение введенного пароля с исходным
7	Пароль верный
8	Пароль неверный
9	Подан сигнал открыть дверь без ввода пароля
10	Состояние ожидание нажатия ручки двери
11	Дверь закрыта
12	Дверь открыта
13	Замок открыт, ожидание открытия двери
14	Ожидание закрытия двери

Таблица 2

Описание переходов СП

Номер перехода	Интерпретация
1	Сравнение паролей
2	Начат ввод пароля
3	Включение тревожной сигнализации
4	Установка исходного состояния
5	Переход в состояние ожидания нажатия ручки двери
6	Установка исходного состояния
7	Переход в состояние ожидания закрытия ручки двери
8	Закрытие замка
9	Ожидание закрытия двери
10	Переход/установка исходного состояния

Разработка программных модулей была осуществлена под операционную систему Windows XP и выше на базе платформы Microsoft .NET Framework 4.0. на языке C# в среде разработки Microsoft Visual Studio 2010 с использование технологии .Net, WPF. Объем программного кода насчитывает 853 строк. Размер исполняемых файлов 75 КБ.

На рис. 5. представлена архитектура разработанного прототипа.

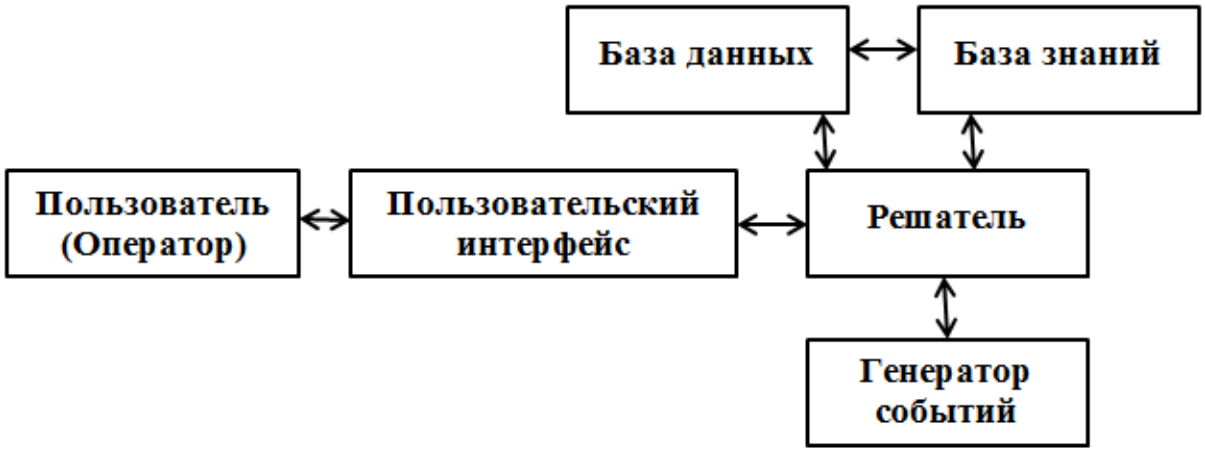

Рис. 5. Архитектура разработанного прототипа

На основании текущего состояния системы генерируются возможные события. Их наступление подтверждается пользователем-оператором и переносится в базу данных. На

основании состояния системы, полученной из базы данных, и базы знаний, заложенной в память программы, решатель автоматически выполняет переход в новое состояние.

В задачи пользовательского интерфейса входит отображение состояния системы, а также уведомление о произошедших в ней изменениях.

Оператор задает начальные настройки системы и производит запуск с помощью кнопки «Старт». Приложение эмулирует различные действия, такие как: ввод пароля, правильность ввода пароля, нажатие ручки двери, открытие и закрытие двери. Система выдает запросы оператору о справедливости тех или иных действий, например: «Нажата ли ручка двери?».

Автоматически происходит обработка переходов и изменения состояния системы. Если система пришла в исходное состояние будет предложен запрос на повторный запуск. В случае включения тревожной сигнализации система придет в режим ожидания ответа от оператора о переводе системы в начальное состояние.

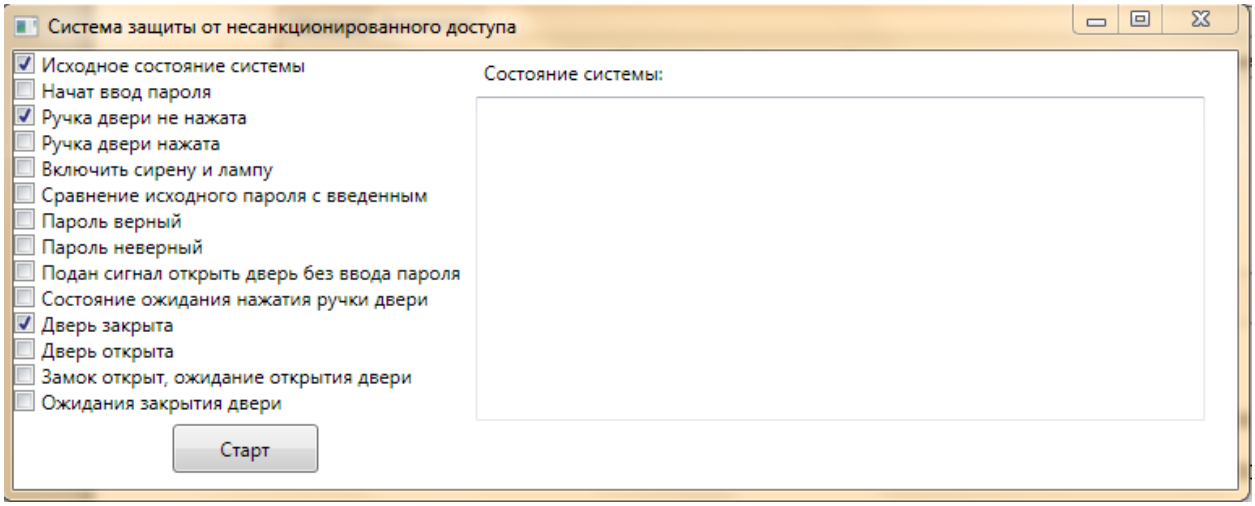

Рис. 6. Графический интерфейс программы

Разработанная система была протестирована на основных, наиболее встречающихся случаях: корректное прохождение в зону, попытка входа в зону при вводе неправильного пароля, прохождение в зону при подаче сигнала «Открыть дверь без ввода пароля».

Задачи, поставленные в рамках выполнения работы, полностью выполнены. Основные результаты:

1. Рассмотрены возможности продукционного подхода в СОЗ и ПЗ с помощью продукционных правил, структурированных на основе математического аппарата СП. В качестве предметной области выбрана система защиты от

несанкционированного доступа. Для описания динамики процесса управления был использован алгоритм на основе рекуррентных уравнений с использованием матриц входов и выходов.

2. Описана программная реализация приложения, использующего разработанный метод.
3. Приведена архитектура главного модуля системы.
4. Подробно описана работа пользовательского интерфейса реализованной системы.
5. Проведено тестирование разработанных программных средств.

Список литературы

1. Аверкин А.Н., Гаазе-Рапопорт М.Г., Поспелов Д.А. Толковый словарь по искусственному интеллекту. М.: Радио и связь, 1992, 255 с.
1. Плотников В.Н., Суханов В.А. Системы, основанные на знаниях. М.: Изд-во МГТУ им. Н.Э. Баумана, 1995. 89 с.
2. Питерсон Дж. Теория сетей Петри и моделирование систем: пер. с англ. М.:Мир, 1984.264с. [James L. Peterson - Petri net theory and the modeling of systems. Prentice-Hall, 1981, 241 p.]
3. Деменков Н.П., Лобусов Е.С., Панин Е.Д., Суханов В.А. Технология реализации компьютерных систем управления на базе структурно-программируемых контроллеров: учебное пособие по курсу «Управление в технических системах» / под ред. К.А. Пупкова. В 3 ч. Ч. 1. М.: Изд-во МГТУ, 1995. 100 с.
4. Гаврилова Т. А., Хорошевский В.Ф. Базы знаний интеллектуальных систем: учебник. СПб.: Питер, 2000. 384 с.
5. Джозеф Джарратано, Гари Райли. Экспертные системы: принципы разработки и программирование: пер. с англ. М.: Издательский дом «Вильямс», 2007. 1152 с. [Joseph Giarratano and Gary Riley - Expert Systems: Principles and Programming 4e, Thomson Course Technology, 2005, 842 p.]