Система классификации химических проб

Г.Т. Маракаева

1. Введение

В данной статье описывается система классификации КХП (Классификатор химических проб), разработанная и реализованная в Московском университете для классификации химических проб. Задача системы классификации состоит в определении класса пробы по заданным значениям атрибутов с использованием ранее накопленных экспериментальных знаний и знаний эксперта. Для классификации химических проб был выбран алгоритм на основе разделяющих функций [1], подробно описанный в работах [2, 3].

Для ускорения классификации в системе предусмотрена возможность использования дерева классов, соответствующих качественным признакам исследуемых проб, которое задается экспертом во время обучения системы. В случае, когда указанное дерево классов определено, его использование позволяет выделить для исследуемой пробы поддерево (подмножество) потенциальных классов и строить разделяющие функции только для выделенного подмножества классов.

Система состоит из набора подсистем, которые программируются на основе набора прикладных интерфейсов (API). Эти интерфейсы используются для реализации новых подсистем системы КХП или для модификации существующих подсистем. Процесс приспособления системы к конкретной задаче классификации называется настройкой системы. Настройка системы позволяет менять свойства системы в достаточно широких пределах. Она осуществляется системным программистом – администратором системы.

Помимо настройки в системе предусмотрен процесс обучения системы, проводимый экспертом. Эксперт вводит в систему обучающую выборку (набор объектов с уже известными классами), на основе которой строится классификационная модель. В текущей версии системы классификация осуществляется с помощью разделяющих функций, так что построение модели сводится к вычислению разделяющих функций по обучающей выборке. Разделяющие функции по набору значений атрибутов объекта выдают ответ о его принадлежности к соответствующему классу.

После обучения системы можно классифицировать объекты неизвестного класса с помощью реализованного в системе Классификатора и получать результат классификации с помощью подсистемы вывода данных.

Поскольку в большинстве современных информационных систем в качестве промежуточного формата представления данных используется язык XML, используемый также в качестве стандарта для интеграции различных систем, то он был для представления данных описываемой системы. Это делает подсистемы ввода и вывода универсальными по данным.

Статья имеет следующую структуру. В разделе 2 описывается архитектура системы – набор интерфейсов и подсистемы, реализованные на базе описанных интерфейсов. В разделе 3 описана подсистема ввода данных, позволяющая получать исходные данные для классификации как из внешних систем, так и с помощью ручного ввода данных в систему. Подсистема обучения, в которой производится построение разделяющих функций, обсуждается в разделе 4. Раздел 5 содержит описание подсистемы классификации для определения класса пробы. Результат классификации выводится с помощью подсистемы вывода, описанной в разделе 6. В разделе 7 рассматриваются представления данных в формате XML. Раздел 8 посвящен требованиям к аппаратуре, необходимой для нормальной эксплуатации системы, а также составу предустановленного программного обеспечения. В заключительном разделе 9 приводятся статистические данные, полученные в ходе эксплуатации системы для классификации химических проб в лаборатории химического предприятия.

2. Архитектура системы

Система состоит из следующих подсистем: классификатор, подсистема обучения, подсистема ввода и подсистема вывода (Рис. 1).

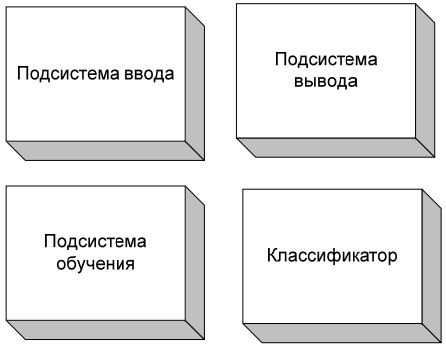

Рис.1. Подсистемы, входящие в состав системы

Таким образом, обеспечивается универсальность построенной системы для решения различного рода задач классификации. Меняя подсистемы, можно добиваться специализации системы для какой-либо конкретной области. Интерфейсы системы, обеспечивающие возможность модификации ее подсистем и построения новых подсистем, представлены на Рис. 2.

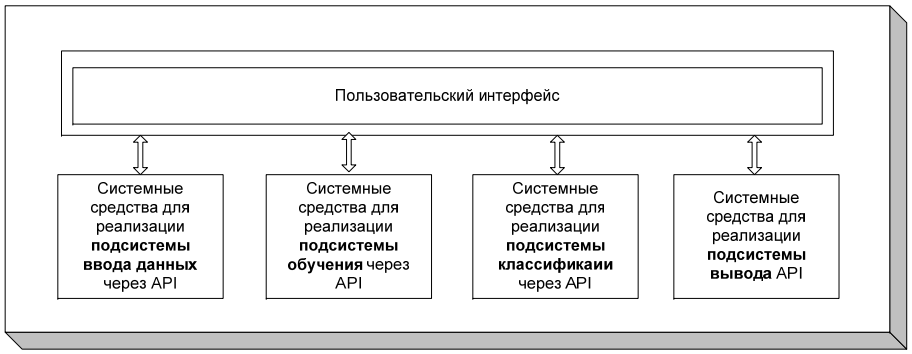

Рис. 2. API и GUI для реализации подсистем

В системе определены процессы настройки, обучения и, собственно, классификации. Настройка системы, выполняемая системным программистом, позволяет изменить алгоритм классификации, и на его основе с использованием интерфейсов программируются подсистемы обучения и классификации.

Обучение системы производится экспертом, который определяет форматы ввода в систему обучающей выборки и вывода результатов. При необходимости эксперт обращается к системному программисту для реализации интерфейсов с приборами.

Для обеспечения модификации указанных подсистем при изменении требований к системе классификации разработан и реализован набор прикладных интерфейсов.

В системе предусмотрены следующие интерфейсы:

1. Интерфейсы для реализации и модификации подсистемы ввода данных:
    a. задание объектов в системе через пользовательский интерфейс;
    b. опции настройки пользовательского интерфейса;
    c. загрузка объектов в систему из внешнего файла;
    d. выгрузка объектов из системы во внешний файл;
    e. загрузка параметров из внешнего устройства в поле объекта;
    f. задание в системе дерева классов;

    g. добавление в системе узлов дерева классов;
    h. добавление селекторов в дереве классов;
    i. задание объекта неизвестного класса.
2. Интерфейсы для реализации и модификации подсистемы обучения:
    a. загрузка в подсистему обучения обучающей выборки;
    b. загрузка в подсистему обучения дерева классов;
    c. выгрузка поддерева дерева классов по заданным параметрам и селекторам переходов между узлами дерева;
    d. выгрузка результата обучения;
    e. интерфейсные выходы в подсистему ввода данных (необходимо для интерактивных алгоритмов классификации);
    f. выгрузка дерева классов.
3. Интерфейсы для реализации и модификации подсистемы классификации:
    a. загрузка в подсистему классификации дерева классов;
    b. загрузка в подсистему классификации результатов обучения;
    c. загрузка объекта неизвестного класса;
    d. интерфейсные выходы в подсистему ввода данных (необходимо для интерактивных алгоритмов классификации);
    e. выгрузка результата классификации;
    f. добавление объекта в обучающую выборку.
4. Интерфейсы для реализации и модификации подсистемы вывода данных:
    a. опции настройки пользовательского интерфейса;
    b. выгрузка объектов из системы во внешний файл;
    c. получение результата классификации.

На рис. 3 в качестве примера приведено полное описание интерфейса «Выгрузка поддерева дерева классов по заданным параметрам и селекторам переходов между узлами дерева».

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Tsa.Opc
{
  public interface ITsaOpcDataSource
  {
    /// <summary>
    /// Функция возвращает массив дочерних элементов для элемента с
полным именем fullName.
    /// Корневой элемент один и его имя "".
    /// При передаче fullName равного null, должен возвращатся массив,
    /// состояший из одного корневого элемента.
```

```
    /// </summary>
    IBrowseTreeSpaceItem[] Browse(string fullName);
    /// <summary>
    /// Функция возвращает сам элемент с полным именем fullName.
    /// Корневой элемент один и его имя "".
    /// </summary>
    IBrowseTreeSpaceItem BrowseItem(string fullName);
    /// <summary>
    /// Получить данные по признаку fullname.
    /// Метод должен вернуть значения, которые входят в интервал
[startTime:endtime] плюс одно
    /// значение, которое "ниже", чем startTime, и одно значение,
которое "выше", чем endTime
    /// </summary>
      }

  public interface IBrowseTreeSpaceItem
  {
    string Name { get;}
    string FullName { get;}

    IBrowseItemProperty[] Properties { get;}
  }

  public interface IBrowseItemProperty
  {
    Type ValueType { get;}
    int PropertyId { get;}
    string Name { get;}
    string Description { get;}
    object Value { get;}
  }
}
```

Рис. 3. Пример интерфейса «Выгрузка поддерева дерева классов по заданным параметрам и селекторам переходов между узлами дерева».

3. Подсистема ввода данных

Подсистема ввода, как и подсистема вывода, не связана с выбранным способом классификации. Основная цель подсистемы ввода – обеспечить ввод в систему данных обучающей выборки, экспертных знаний в виде дерева

классов и объекта неизвестного класса. Подсистема ввода в описываемой реализации позволяет вводить данные четырьмя различными способами: ввод дерева классов, ввод данных обучающей выборки и объекта неизвестного класса через пользовательский интерфейс системы, загрузка обучающей выборки из внешних источников и получение данных по значениям признаков для обучающей выборки из внешних приборов (Рис. 4).

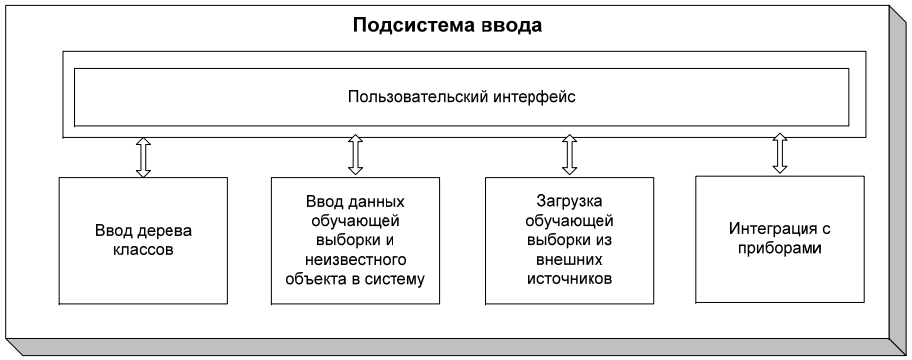

Рис. 4. Подсистема ввода.

Представление экспертных знаний в системе реализовано в виде дерева классов. При исключении подсистемы обучения общая система будет представлять работу экспертной системы на основе структуры классов, заложенной в систему экспертом, и, таким образом, система будет являться только экспертной. При построении классификационной модели эксперт на основе своих знаний задает иерархию классов в виде дерева. С помощью этого механизма достигается сокращение числа операций при классификации. Задание дерева происходит следующим образом: каждому узлу соответствует класс, при этом родительский узел обозначен классом – обобщением своих дочерних узлов. Корневому узлу приписываются все классы классового пространства. Ребра дерева могут содержать селекторы – условия перехода от обобщенного класса к уточненному.

Основная работа лаборанта заключается в проведении экспериментов и фиксации результатов. Поэтому в подсистеме ввода предусмотрен журнал проведения экспериментов. Внешне он очень похож на свой бумажный аналог, что позволяет лаборантам начать пользоваться системой без какой-либо длительной подготовки. Но, несмотря на свою внешнюю простоту, в электронный журнал встроено множество функций:

- структурирование всех журналов в удобную для использования иерархию;
- регистрация проб, поступивших на испытание;

- выбор значения параметра из заранее заданного списка;
- автоматический расчет параметров по заданным формулам (математические формулы – результатом является число, логические формулы – результатом является, например, “соответствует/не соответствует норме”);
- выполнение расчетов с заданной точностью;
- сигнализация (например, выделение красным цветом) значений, не попадающих в допустимый диапазон;
- быстрый поиск нужного эксперимента, по любому параметру;
- быстрая сортировка записей экспериментов по любому параметру (по возрастанию/убыванию);
- выполнение параллельных испытаний с последующей консолидацией результатов;
- поверка приборов;
- хранение информации о том, кто выполнил эксперимент;
- выгрузка/загрузка одного журнала (или всех журналов) с возможностью отправки по электронной почте или записью на внешний носитель;
- возможность ввода и ведения атрибутов журнала (ФИО ответственного за методику сотрудника, список приборов, используемых в определении показателей для журнала и т.д.);
- разделение прав доступа пользователей на чтение/запись/изменение данных в электронном журнале;
- возможность настройки ролей пользователей (например, выделяется роль лаборанта, к которой приписываются все лаборанты лаборатории; таким образом, не надо прописывать каждому пользователю права отдельно; это особенно актуально для больших лабораторий, в которых используется много методик).

Ручной ввод данных в систему осуществляется по заранее заданным шаблонам представления объектов (Рис. 5.). Чтобы избежать ошибок ввода для обучающей выборки (Рис. 6), в систему встроен механизм проверок данных на правильность и целостность (например, если агрегатное состояние пробы питьевой воды выбрано как «лед», то температура пробы не может быть +5ºС). Некоторые признаки объектов могут рассчитываться по формулам на основе введенных значений других признаков, полученных экспериментальным путем.

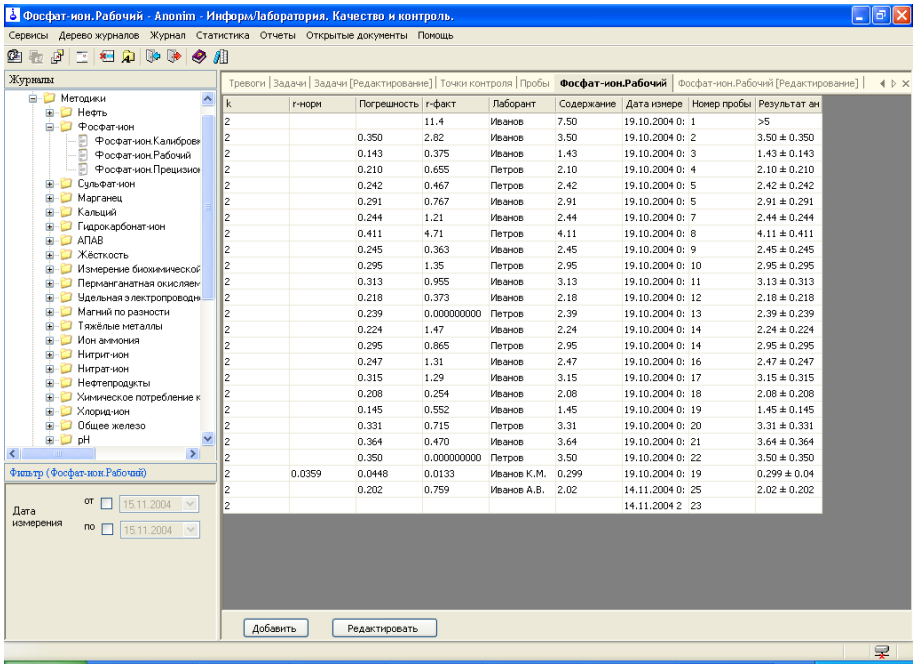

Рис. 5. Экранная форма с введенными вручную данными.

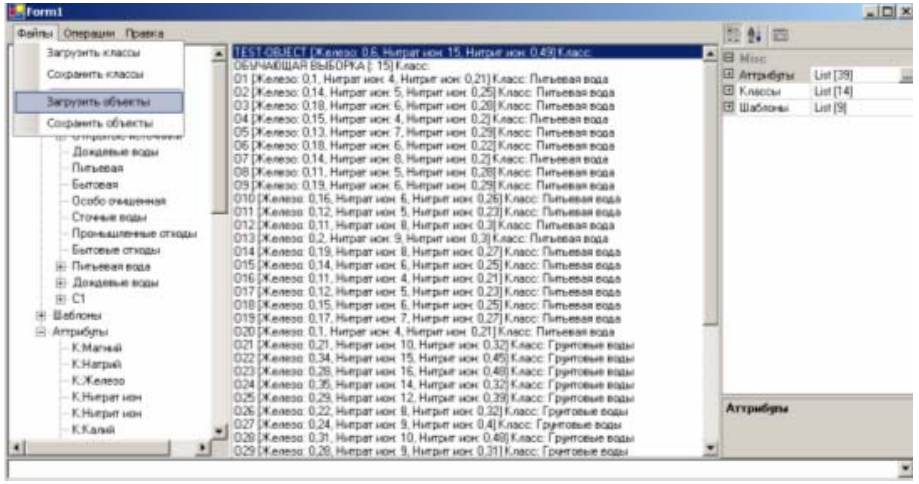

Рис. 6. Экранная форма работы с внешними данными: загрузка и выгрузка обучающей выборки.

4. Подсистема обучения

Основная задача подсистемы обучения – построить разделяющие функции для каждого класса, представленного объектами обучающей выборки (рис. 7).

Рис. 7. Подсистема обучения.

Подробно алгоритм обучения описывается в [1, 2, 3].

Для пользователя системы работа подсистемы обучения практически не видна. Пользователь запускает функцию «Обучение», и по окончанию обучения системы он видит в окне отчетов работы системы сообщение об окончании обучения (Рис. 8).

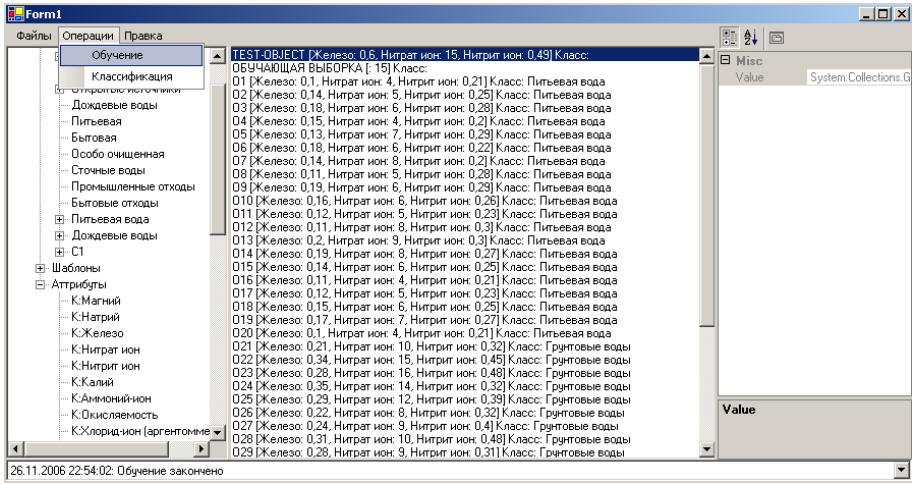

Рис. 8. Запуск и выполнение обучения.

5. Классификатор

Классификатор представляет собой подсистему, реализующую применение экспертных и экспериментальных знаний для основной цели системы – классификации пробы неизвестного класса (Рис. 9).

Подробно алгоритм классификации описывается в [1, 2, 3].

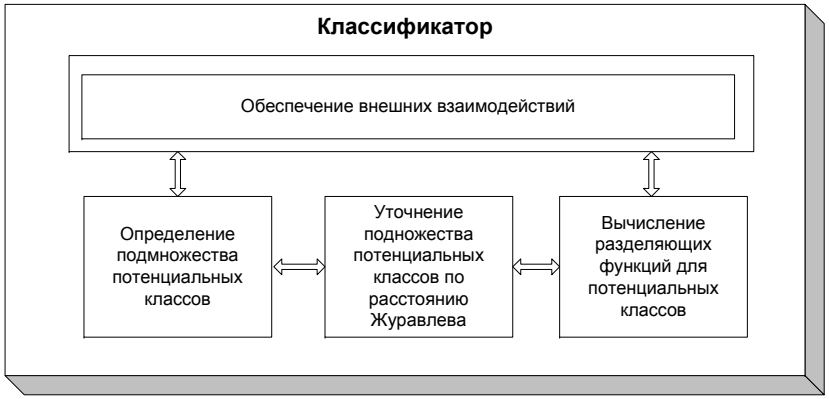

Рис. 9. Подсистема Классификатор.

Для начала работы с классификатором пользователь должен ввести объект, задать значения признаков и оставить класс объекта пустым, затем запустить операцию классификации и дождаться результата в нижней части окна системы (Рис. 10).

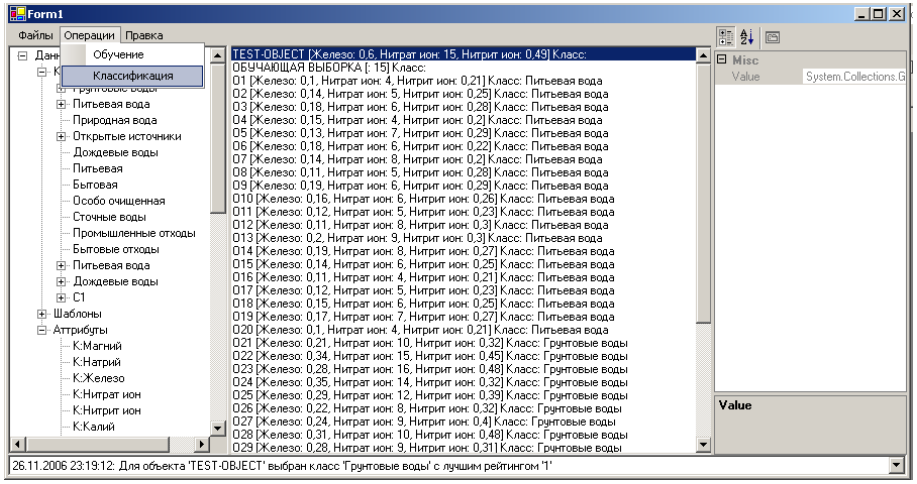

Рис. 10. Классификация неизвестной пробы.

6. Подсистема вывода данных

Подсистема вывода данных, как и подсистема ввода данных, не связана с выбранным алгоритмом классификации и другими подсистемами системы, что позволяет в случае необходимости подменять эту подсистему на другую и делает систему универсальной по выводу данных.

Подсистема вывода данных, реализованная в описываемой работе, позволяет выводить введенные данные о пробах и результаты проведения классификации в заранее определенном требуемом формате, в соответствии со стандартами, принятыми в организации, которая эксплуатирует систему. В библиотеке ядра системы содержатся функции ввода-вывода данных, с помощью которых реализована подсистема.

Выгрузка данных о пробах производится в документ, называемый паспортом качества продукции, так как анализ проб очень часто проводится для доказательства соответствия пробы некоторым стандартам.

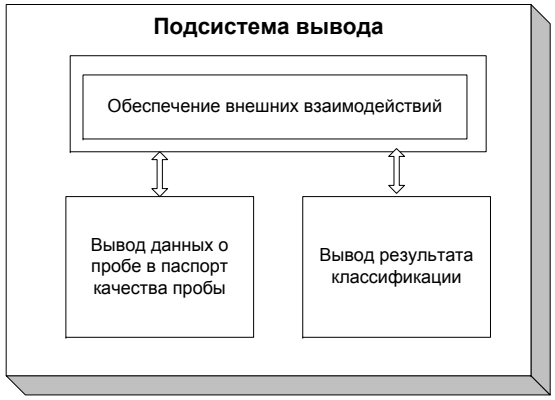

Рис. 11. Подсистема вывода.

Выдача результата классификации проводится в менее формализованном виде – в окне результатов выводится название пробы и полученное при классификации имя класса. Такое простое представление результата связано с тем, что результат классификации служит лишь для того, чтобы помочь пользователю при определении класса, но для официального подтверждения необходимо проводить эксперименты по всем значащим признакам пробы (Рис. 11).

7. Представление данных

На Рис. 12 представлена схема ввода данных в систему. Подробно ввод данных описан в разд. 3.

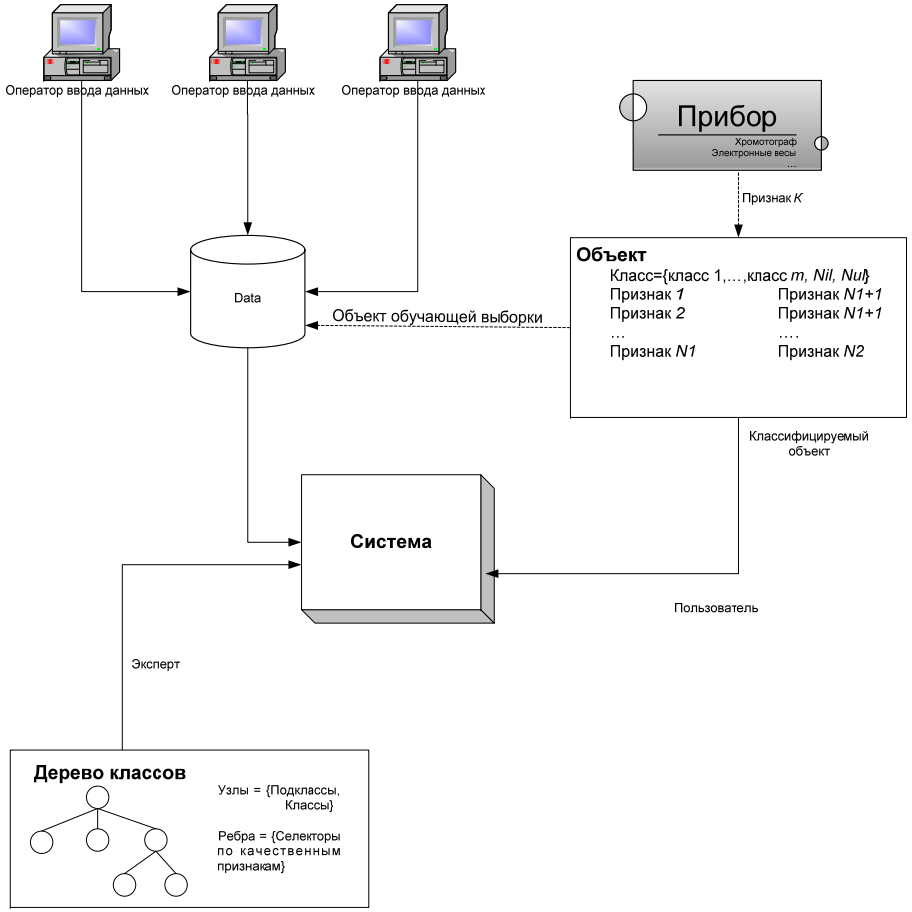

Рис. 12. Модель поступления данных в систему.

```
<!ELEMENT object (name, class?, attribute*)>
<!ELEMENT attribute (name, value+)>
<!ELEMENT name (#ID)>
<!ELEMENT value (#PCDATA)>
```

Рис. 13. Шаблон для хранения информации об объектах в формате DTD.

Поскольку в большинстве современных информационных систем в качестве промежуточного формата представления данных используется язык XML, используемый также в качестве стандарта для интеграции различных систем,

то он был использован для представления данных описываемой системы. Это делает подсистемы ввода и вывода универсальными по данным.

Для наиболее эффективного хранения информации, а также для оптимальной интеграции с разнородными источниками данных был разработан специальный шаблон хранения информации. Для описания формата хранения был использован стандарт DTD – Document Type Definition (Рис. 13). Данный стандарт широко известен и часто используется при работе со средствами XML, несмотря на развитие новых способов описания метаданных XML. В рамках нашей системы допустимым признается корректно сформированный XML файл, не противоречащий данному определению типа документа.

```
<?xml version="1.0" encoding="utf-8" ?>
- <Data xmlns:xsi="http:// XMLSchema-instance"
      xmlns:xsd="http://XMLSchema">
- <Class>
  <Name>Грунтовые воды</Name>
    </Class>
- <Class>
  <Name>Питьевая вода</Name>
    </Class>
- <Class>
  <Name>Природная вода</Name>
    </Class>
- <Class>
  <Name>Открытые источники</Name>
    </Class>
- <Class>
  <Name>Дождевые воды</Name>
    </Class>
- <Class>
  <Name>Питьевая</Name>
    </Class>
- <Class>
  <Name>Особо очищенная</Name>
    </Class>
- <Class>
  <Name>Сточные воды</Name>
    </Class>
- <Class>
  <Name>Промышленные отходы</Name>
    </Class>
- <Class>
```



```
  <Name>Бытовые отходы</Name>
    </Class>
- <Class>
  <Name>Питьевая вода</Name>
    </Class>
- <Class>
  <Name>Дождевые воды</Name>
    </Class>
- <Class>
  <Name>C1</Name>
    </Class>
    </Data>
```

Рис. 14. Пример описания словаря классов в формате XML.

Контейнер верхнего уровня – это объект, который в рамках системы может быть одним из различных функциональных элементов в зависимости от того, какому классу он принадлежит. Класс принадлежности объекта определяется в соответствии с вложенным контейнером. Единицей, хранящей информацию, является набор атрибутов, которые, согласно приведенному определению типа документа, могут быть произвольного типа и в любом количестве.

На Рис. 14 представлен пример этого формата – XML-файл, описывающий словарь классов.

8. Технические характеристики системы

Система реализована на языке C# в среде VisualStudio. Данные в системе представлены в формате XML, что обеспечивает дополнительную универсальность в использовании данных из каких-либо внешних систем.

Реализация системы состоит из приблизительно 5500 строк, из которых 3500 строк – инфраструктура системы в ядре, остальной код – реализация системы для решения задачи классификации химических проб.

Требования к аппаратуре: Для работы с системой рекомендуется следующая конфигурация рабочей станции: Pentium III, тактовая частота не менее 500МГц, свободное место на диске (HDD) не менее 500Мб, оперативная память не менее 256Мб.

Минимальная конфигурация: Celeron, тактовая частота 200МГц, свободное место на диске 150Мб, оперативная память 64Мб.

Требования к системному ПО: Для работы оператора с системой рекомендуется следующая конфигурация программного обеспечения (ПО) для рабочей станции: Microsoft Windows 2000, Microsoft Internet Explorer 5.5, Microsoft Office 2003, XP.

Минимальная конфигурация ПО для рабочей станции: Microsoft Windows 98, Microsoft Internet Explorer 4.0.

9. Результаты экспериментальных расчетов

В Табл. 1 приведены некоторые результаты проведения экспериментов, полученные в ходе тестирования системы.

количество классов	число объектов обучающей выборки	количество количественных признаков	число объектов обучающей выборки для 3-х классов	число объектов обучающей выборки для 5-х классов	число объектов обучающей выборки для 7-х классов
0	0	0	0	0	0
1	0	1	14	67	70
2	8	2	24	87	84
3	16	3	22	86	92
4	25	4	25	94	120
5	39	5	39	89	140
6	59	6	41	110	144
7	87	7	38	98	128
8	120	8	35	101	129
9	160	9	34	102	148
10	220	10	36	89	133
11	269	11	32	92	152
12	314	12	33	95	146
13	420	13	34	101	178
14	560				
15	700				

Таблица 1. Результаты проведения экспериментов

Приемлемое число объектов обучающей выборки (при условии нормального распределения значений атрибутов) в зависимости от количества классов приведено на Рис. 15.

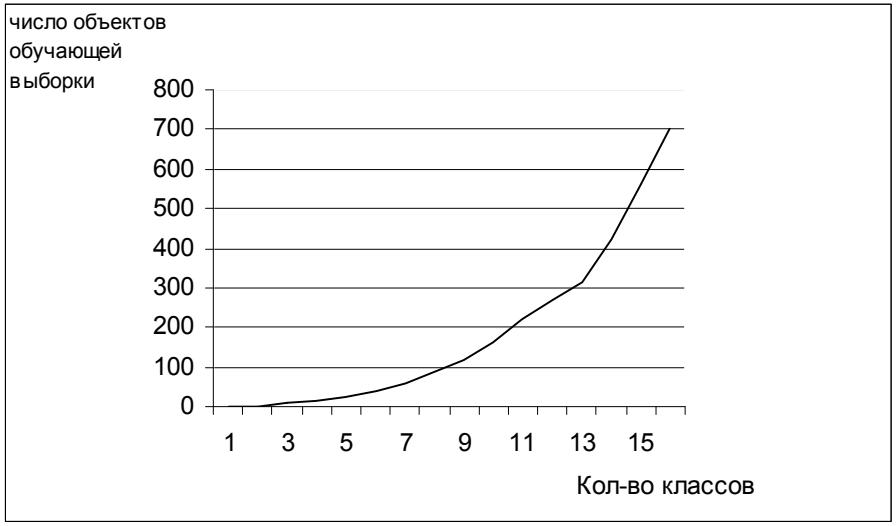

Рис. 15. График зависимости приемлемого числа данных объектов обучающей выборки от числа классов.

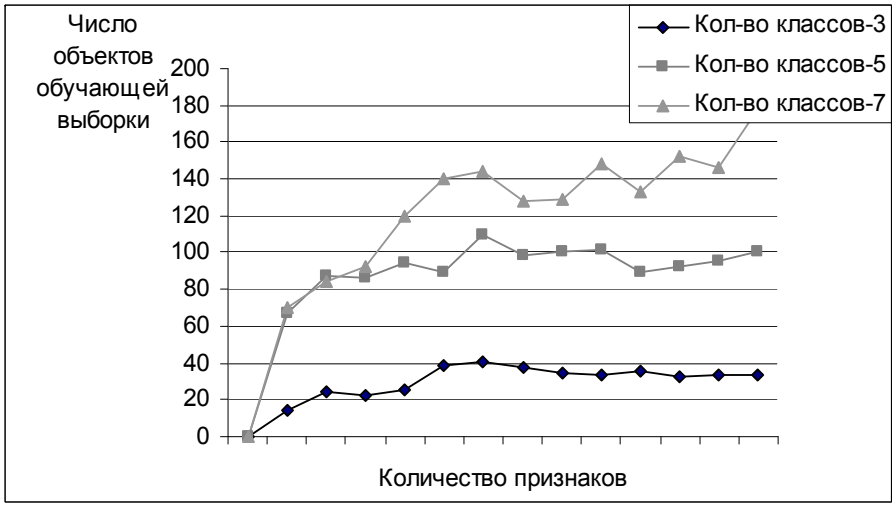

Рис. 16. График зависимости приемлемого числа данных объектов обучающей выборки от количества признаков.

Приемлемое число объектов обучающей выборки (при условии нормального распределения значений атрибутов) в зависимости от количества признаков для разного числа классов приведено на рис. 16.

Литература

[1] Маракаева Г.Т. "Применение методов выявления закономерностей для классификации химических соединений", Сборник статей ИСП РАН, 2006

[2] Маракаева Г.Т. "Использование подходов Data Mining в развитии систем ЛИМС", ежемесячный научно-технический и производственный журнал "Автоматизация в промышленности", №12, 2006

[3] Маракаева Г.Т. "Обоснование правильности работы алгоритма классификации", тезисы к докладу на международной московской конференции Ломоносов 2006, МГУ, 2006.