STUDENT DEVELOPER TEAM
«THE NEXT AGE TEAM»

I. Brief Introduction

«The Next Age Team» is a developer team created in 2003. Initially it was a group of students of Computer Science School who were in training for taking part in student programming contests and participated in those competitions. After some time the group expanded the area of its activity. For one thing, participating in the competitions allowed to the students to become a real team, not a simple group of people gathering just for some hours a week. Being a real team means operating like a single one, that allows us to solve quite complicated tasks and problems and create software systems of different complexity levels and for different applied fields. In the second place, taking part in the contests improved our programming level significantly. The specific characters of the contests caused the fact that we had to study a lot of algorithms and development technologies which are over and above the standard educational program. Also the extreme conditions, happening at the competitions, caused increase in quality of coding, testing, debugging and some other aspects important for applied programming also.

Now our team consists of 15 members and we can say about the following directions of our activity:

1. Development of complex and science intensive program systems using different algorithms:
2. Development of commercial software systems ordered by enterprises;
3. Our own research and, if it is possible, development program systems based on this research results.

In the period of 2003 – 2006 we designed and implemented 33 projects, some of them in collaboration with other programming teams. Several members of the team took part in Intel Summer School program (Intel Nizhniy Novgorod Lab, Russia) and Samsung Summer Internship program (Samsung Digital Complex, Suwon, Republic of Korea). Also we have Microsoft partner students in the team. For our work we use the listed languages, platforms and technologies:

- **Languages:** C++ (Visual C++), C#, Java, Visual Basic, Assembler
- **Platforms** and **devices:** Windows, Web, .NET, PDAs, mobile phones, network programming, other devices controlled with OS Windows, Linux
- **Databases:** SQL, MySQL, OleDb, technologies ODBC, ADO, ADO.NET
- **Web-design** and **Web-programming**
- Parallel programming, mathematical algorithms, modelling algorithms, AI algorithms, Data Mining etc.

We would be glad if this information will interest developers, managers, researchers, potential partners and customers. Our contact information is below:

- **Kryzhanovsky Dmitry**, Team Leader,
  E-mail: dmitry.kryzhanovsky@gmail.com,
  Phone: +7-905-332-84-87, +7-909-906-55-83

- **Girkin Mikhail**, Senior Developer,
  E-mail: mikegirkin@gmail.com,
  Phone: +7-927-509-24-35

You can visit our web-site http://www.thenextage.com to find more detailed information.

II. The Projects Implemented in 2003 – 2006

The brief list of the projects implemented in 2003 – 2006 is here below. The projects developped in collaboration with other programming teams marked with an asterix *:

1. **Web-sites:** 9 Web-sites of different complexity, form simple calling-card sites to Web-systems of building optimal routes between the specified points; the most complicated projects in this sphere is SIMON Redesign worked by the order of Teralogix, a company of the USA (it will be available on https://teralogix.simonasp.com soon), this system is designed for managing IT projects.

2. **Educational and Human Service Software:**
   - System for automated calculation of students' rating
   - Distributed system to conduct and control programming contests
   - * 2 systems for psychological testing of the staff, on demand of employment agencies

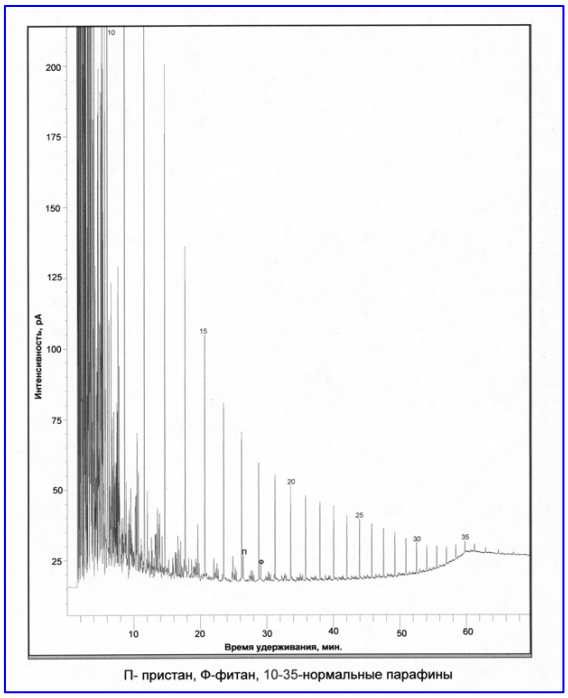
3. **Serious Science Intensive and Industrial Projects:**
   - SynOIL – a software system to analyze characteristics of oil and gas samples, developped on demand of LUKOIL, the version 1.0 in 2005 and 2.0 in 2006
   - Automated system to control tank-robot battle
   - 3 systems to find hidden dependencies in data arrays
   - * System to model the work of trucking industry
   - * Control program system for pipe diagnostic robot (by request of Gazprom)

- Expert support system for making decisions
- * Modules for CAD-system «Smart Application for Engineering Analysis», developped in the course of international university project
- * 2 modules for ANSYS for stress analyze of corks in gas globe vales

4. **Commercial Orders:**
   - Program for data conversion and administration in ACNielsen, marketing research company
   - A common core of automatic control system for enterprises
   - Mobile commercial agent
   - System to optimize losses because of slitting metal and wooden sheets
   - System to design window and door constructions

5. **Other Projects:**
   - Emulator of i8051
   - SMTP / POP3 mail server
   - CryptoAPI user interface

2.1. A Software System to Analyze Phase and Genetic Characteristics of Reservoir Fluids



Pict. 1. Chromatogram

Experts, who search undiscovered fields of oil and gas, use phase and genetic characteristics of the oil samples for their work. Phase characteristics show us what conditions the explored oil reservoir fluid is in (temperature, pressure, phase consistent: liquid oil, gas, bitumen and so on), and the genetic characteristics reflect the process of genesis of the oil that allows to the experts to restore the picture of the explored sample evolution.

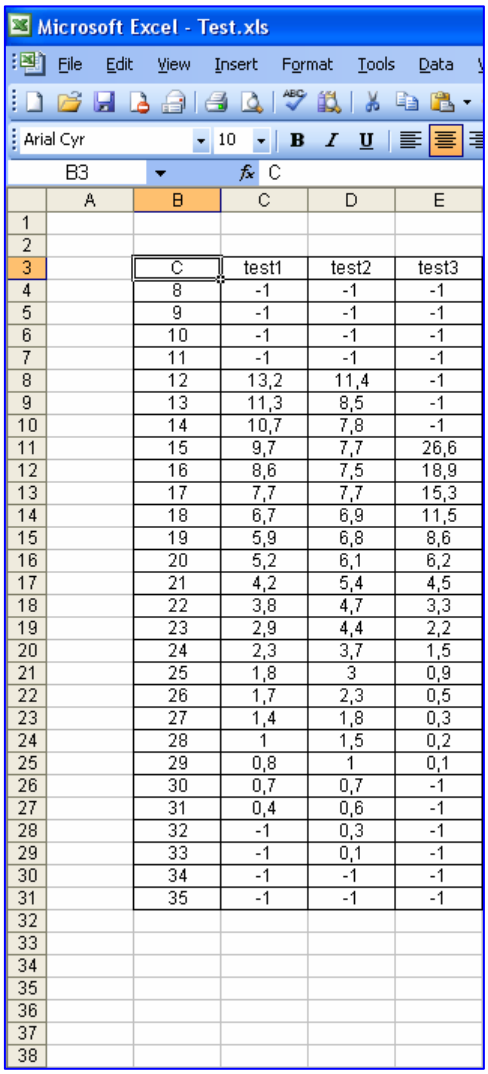
The source information about oils is chromatograms that depict how much (in percents) the explored sample contains different hydrocarbons. So it can be presented like a diagram on the plane where the axis OX contains the numbers of carbon atoms in hydrocarbon molecules, and the axis OY contains percentage corre-

sponding to these hydrocarbons.

Such chromatograms are produced by special equipment, and after one obtaining the diagrams the experts start to analyze phase and genetic characteristics of the investigated sample. The hydrocarbon distribution, presented on the picture, defines the industrial quality of the oil (what product we can produce from it and what features it will have) and also allows to restore the evolution of the oil. The latter is very important since after restoring the conditions of oil genesis we can predict whether other unexplored fields are around and where we should find them with the higher probability.

The difficulty happening here is to divide the original sample into the phase this sample consists of. The source chromatogram shows only the sample itself, not its phases separately. But it's important to know the percentage of each phase to analyze the oil. There are several methods to do it, and one of them, developped in a laboratory of LUKOIL was implemented in the software system we are describing here.

In general the system works in the following way. A user opens an Excel table containing the information about hydrocarbon distribution (the rows are hydrocarbons, the columns are different wells, and cells show percentages). The user selects the columns (wells) to investigate and after that the process of modelling runs.



	A	B	C	D	E
1					
2					
3		C	test1	test2	test3
4		8	-1	-1	-1
5		9	-1	-1	-1
6		10	-1	-1	-1
7		11	-1	-1	-1
8		12	13,2	11,4	-1
9		13	11,3	8,5	-1
10		14	10,7	7,8	-1
11		15	9,7	7,7	26,6
12		16	8,6	7,5	18,9
13		17	7,7	7,7	15,3
14		18	6,7	6,9	11,5
15		19	5,9	6,8	8,6
16		20	5,2	6,1	6,2
17		21	4,2	5,4	4,5
18		22	3,8	4,7	3,3
19		23	2,9	4,4	2,2
20		24	2,3	3,7	1,5
21		25	1,8	3	0,9
22		26	1,7	2,3	0,5
23		27	1,4	1,8	0,3
24		28	1	1,5	0,2
25		29	0,8	1	0,1
26		30	0,7	0,7	-1
27		31	0,4	0,6	-1
28		32	-1	0,3	-1
29		33	-1	0,1	-1
30		34	-1	-1	-1
31		35	-1	-1	-1
32					
33					
34					
35					
36					
37					
38					

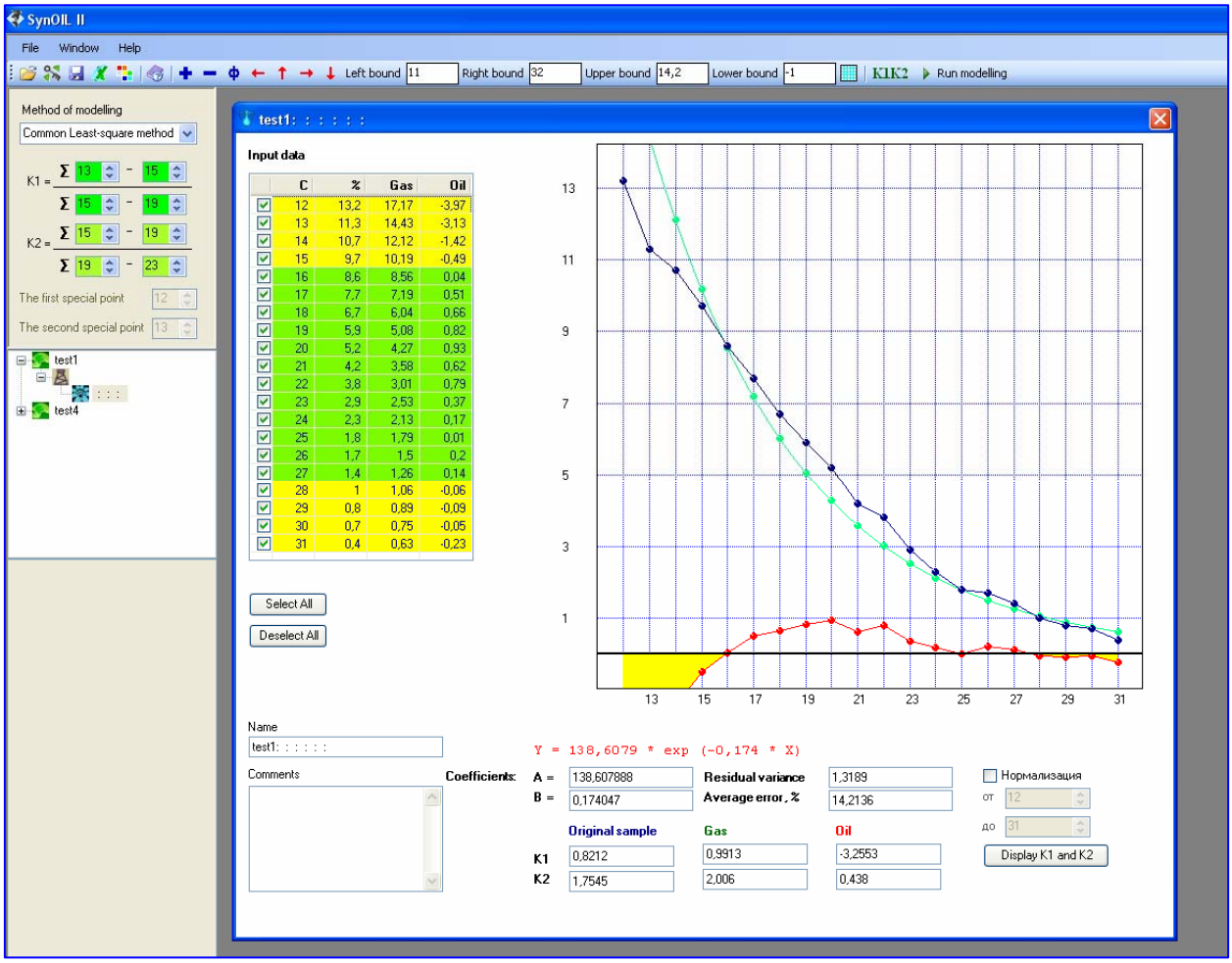
Pict. 2. A source Excel file

On the picture 3 you can see a result of modelling for one sample. Using embedded algorithms the application divides the phases and builds a diagram showing how the phases behave inside the original sample. Since there are many factors influencing the phases inside an oil fluid, the application allows to the user to adjust the built model manually changing the values in the text boxes and other controls.

The application also computes additional characteristics which can help to the experts in analyzing the oil. For example, the coefficients K1 and K2 that show the ratio of percentage light hydrocarbons (in the left part of the chromatogram) and heavy hydrocarbons (in the right part). These coefficients are important since they are very informative for predicting the industry quality of the oil and for restoring a general picture of its evolution.

This application was tested in some laboratories exploring new oil fields (in 2005 the version 1.0 was realized, the version 2.0 was done in 2006), and in future this system will underline a

more complicated system, which will be an expert system to analyze the quality of the oil.



Pict. 3. A result of modelling

2.2. A Software System to Mode the Work of Trucking Industry

This system was developped in the course of investigation concerning the ways and methods to increase in transportation effectiveness (the research was conducted in Motor Transport School). There were selected about 130 different parameters that influence the transportation effectiveness (more detailed, four effectiveness coefficients which shows losses and profits taking place during the transportation).

It is important to emphasize the following two things:

1. The program models not one carting from the sender to the receiver. It models the work of N (N is set by the user) transportation enterprises, senders, receivers, terminal regional centres, railway stations, loading equipment, warehouses and so on. In fact, when the model was being

tested, users set N equal to 200 or even more. So the application models the work of a great part of trucking industry for some regions.

2. And the second one. The modelling method used in the system is a simulation. It means that the application doesn't use the old formulae for modelling. It runs the modelling process in a discrete time and simulates all the parameters and object behaviour as if they operate in a real time. For example, modelling the work for one month of real time took about 9 hours of computer time.

The results obtained on the models built by the application were used in Regional Transport Management and they underlined a PhD thesis.

III. The Projects Being Developped at Present

3.1. A Code Game Challenge Control System

A Code Game Challenge is an informal title for competitions of game strategies that initially took place at the final rounds of ACM ICPC (ACM International Collegiate Programming Contests). You can read about that more detailed on the sites http://icpc.baylor.edu/icpc/ (the official site of ACM ICPC), http://contest.sgu.ru (the official site of South Russia quarter final round) and some other pages.

The idea of the competition is following: the judges provide the competitors with the class interface for interacting between the competitors' units and the server. Each competitor team has 5 hours to realize some game strategy that meets the requirements proposed by the jury and the laws of the world invented by the author of the competition. During the competition the teams have possibility to set on fight their game units with the units of other teams in order to test it. After these 5 hours elapsing, the jury starts to set on fight all game strategies invented and implemented by the teams in order to define the best of them.

And while the fight is running, it is demonstrated on the screen, so it looks like a sport competition (many people like to watch horse race or other races), but instead of real horses, cars and peoples programmed units strike in this fight. And it is a question of decoration whether we choose tanks or spaceships.

Though it can be seemed that it is just a game, the experience has showed that competitors invent very interesting and complicated strategies even within 5 hours, and this competition can stimulate more serious investigations of game strategies (not only battle), multiagent system, artificial intelligence, the management theory and so on.

3.2. A Distributed Software System for Researching Game Strategies (by the Example of Tank-Robot Battles)

This project originates from the previous one. The difference here is in highlighting key points. If the first project is designed for conducting such competitions in our university, the described project has another target – to create a system for researching game strategies and models of cooperative behaviour of intelligence agents.

Pict. 4. A gunroller-robot for game strategy research

Another important difference is the way to code (implement) strategies. If in the first project it is necessary to be a programmer and to know a programming language (C++ and Java are used much more often than the other), the present project is designed for a wider circle of users. So, there will be three ways to code strategies: using a programming language; using a special language for game strategies and robots (such languages exist and they are not designed just for programmers); adjusting the strategy via manipulating its numeric and linguistic parameters.

A tank-robot battle was selected as a demonstration method just for popularization and commercial considerations. The system itself will not be linked with tanks and its target to help in research game strategies, multiagent system, cooperative behaviour and so on using real-time modelling and simulation (that is implemented via our program).

3.3. A Pocket Navigator for Mobile Phones

The project was started in July 2006 and is being developped now. The heart idea is following.

Even in the age of GPS to orientate on place is a problem for many people since only some of them have GPS devices, but the problems of finding a correct way or defining the current location arises for most of us. The list of situations when people (even who orientate on place well) face such geographical problem is quite big. But in our epoch we can find an alternative for GPS, and we suppose that a usual mobile phone can be such an alternative.

Mobile applications, mobile services are common and usual in our time, so there are not fundamental difficulties to use mobile phones for navigation. There have already created many applications and mobile services (provided by cellular phone companies) for defining the current location of the object.

Using the existing technologies and the fact that in the city and on the road we have a limited number of degree of freedom (we can’t go through the walls and, it most likely, that in the city we will go in the streets, not in the yards and so on) we are developping now a system which combines the existing abilities and provide some new. The short list of the main functions is below:

- Defining the current location
- Defining the distance between the current location and specified object (or between some specified objects)
- Estimating the time necessary to get to the specified points
- Compass + monitoring what direction you should go
- Finding the optimal way (optimization criteria can be different and a user can choose them) for walkers, cars, bikers, for people using public conveyances and so on

The result of work can be presented in two forms: graphics (map) and text.

IV. Taking Part in Student Programming Contests

For 31 years ACM has conducted International Collegiate Programming Contest (ICPC), and this movement is getting more and more popular. In some recent years there were organized many other contests using rules and considerations of ACM ICPC, though, of course, ICPC remains the most important, difficult and prestigious programming contest in the world.

You can find more information on the official ACM ICPC site http://icpc.baylor.edu/icpc/. Many people think that student contests is a game, and it is not worth to spent time for it. We can say surely that it is wrong. Participating in these contests helps to increase programming level significantly. Since all these competitions are not for individual programmers but for the teams, it also helps to study to work in the team that is especially important if one wants to take success in career.

We have taken part in such competitions since 2003. During this period we participated four times in South Russia Student Programming Contest taking place in Taganrog Radio Engineering University, where we were able to set our university at the second place in the rating (the first one is of Taganrog University itself).

Participating in ACM ICPC we had, of course, more unassuming results. We took part four times in South Russia quarter final round, and moved from the 24 position in 2003 (among 50 teams) to the 12 position in 2006 (among 65 teams). In 2006 a good fortune smiled us at last and we passed into the next round (Russia and the neighbour countries semi-final, or North-Eastern European semi-final). So ICPC is a difficult thing, but it is worth to work at it. Even if one will not pass into a semi-final or a final, it will help him to become an excellent programmer.