WEB-SYNDIC

Web System for Demonstrating the Syntactic Algorithms for Solving Linear Equations in Nonnegative Integers (Nonnegative Linear Diophantine Equations)

MAINTENANCE DOCUMENT

Department of Computer Science, Petrozavodsk State University, Russia

15th November 2004

Move to Russian version

Contents

1 General

Modification History

Version	Date	Author(s)	Modification	Reason (optional)
1.00 (1.1)	16.07.2003	D. Korzun, Y. Bogoyavlenskiy	Initial version	
1.10 (1.2)	08.08.2003	Y. Bogoyavlenskiy	Customer's corrections	Customer's inspection
1.11 (1.3)	15.08.2003	D. Korzun	Some minor improvements of the text	Senior Manager's inspection

1.1 General: Version 1.00

Nonnegative linear Diophantine equations (NLDE) are linear equations with integer coefficients and with solutions in nonnegative integers. NLDE and their systems are well-known area of mathematical research, which goes back to Ancient Greece. Nowadays the algorithmic issues of this area are the nest of diverse topical problems.

An interesting case is NLDE systems associated with context-free (CF) grammars (ANLDE systems). The theory of ANLDE systems is a subject of recent research at the Department of Computer Science (CS) of Petrozavodsk State University (PetrSU). This theory has roots in the work of Miguel Filgueiras and Ana-Paula Tomás [1] (Universidade do Porto, Portugal). Their results were later developed by Dmitry Korzun (PetrSU) in his PhD [2] with introducing new efficient polynomial algorithms for solving some classes of ANLDE-systems [3]. These algorithms are called “syntactic” because of their grammar-based nature.

Nowadays Web is the standard way to introduce results to whom it may concern. This Project is intended to demonstrate the computational efficiency of the syntactic algorithms for a wide spectrum of researches through a web system. The system should visually, on manually and automatically generated examples demonstrate to a user what are the syntactic algorithms, what computational behavior they have, and how efficiently they work comparing with some available algorithms of other authors.

This Project is a student software engineering (SE) project in the framework of cooperation between the CS Departments of PetrSU and the University of Helsinki (UH). The developers are students of the CS Department of PetrSU. They study course “Software Engineering”. The Project for them is an exercise for applying SE theory in practice. The CS Department of UH helps the PetrSU students to study these SE topics, consults them on the related questions, and makes an expert estimation of the process.

The short name of the Project is **Web-SynDic**. This means that the target system is Web-based; “SynDic” is the word “Syntactic” (the algorithms are syntactic), where “tact” is replaced with “D” according to our great tact to Diophant as a father of NLDE Theory.

1.2 General: Version 1.10

Nonnegative linear Diophantine equations (NLDE) are linear equations with integer coefficients and with solutions in nonnegative integers. NLDE and their systems are well-known area of mathematical research, which goes back to Ancient Greece. Nowadays the algorithmic issues of this area are the nest of diverse topical problems.

An interesting case is NLDE systems associated with context-free (CF) grammars (ANLDE systems). The theory of ANLDE systems is a subject of recent research at the Department of Computer Science (CS) of Petrozavodsk State University (PetrSU). This theory has roots in the work of Miguel Filgueiras and Ana-Paula Tomás [1] (Universidade do Porto, Portugal). Their results were later developed by Dmitry Korzun in his PhD [2] (PetrSU, under scientific supervising of Dr. Yury Bogoyavlenskiy) with introducing new efficient polynomial algorithms for solving some classes of ANLDE-systems [3]. These algorithms are called "syntactic" because of their grammar-based nature.

Nowadays Web is the standard way to introduce results to whom it may concern. This Project is intended to present the syntactic algorithms for wide independent testing and to demonstrate for a wide spectrum of researches the computational efficiency of the algorithms. The target web system should visually, on manually and automatically generated examples present main ideas of the syntactic algorithms, demonstrate to a user what computational behavior they have, and how efficiently they work comparing with some available algorithms of other authors.

This Project is a student software engineering (SE) project in the framework of cooperation between the CS Departments of PetrSU and the University of Helsinki (UH). The developers are students of the CS Department of PetrSU. They study course "Software Engineering". The Project for them is an exercise for applying SE techniques in practice. The CS Department of UH helps the PetrSU students to study these SE topics, consults them on the related questions, and makes an expert estimation of the process.

The short name of the Project is **Web-SynDic**. This means that the target system is Web-based; "SynDic" is the word "Syntactic" (the algorithms are syntactic), where "tact" is replaced with "D" according to our great respect to Diophant as a father of NLDE Theory.

1.3 General: Version 1.11

Nonnegative linear Diophantine equations (NLDE) are linear equations with integer coefficients and with solutions in nonnegative integers. NLDE and their systems are well-known area of mathematical research, which goes back to Ancient Greece. Nowadays the algorithmic issues of this area are the nest of diverse topical problems.

An interesting case is NLDE systems associated with context-free (CF) grammars (ANLDE systems). The theory of ANLDE systems is a subject of recent research at the Department of Computer Science (CS) of the Petrozavodsk State University (PetrSU). This theory has roots in the work of Miguel Filgueiras and Ana-Paula Tomás [1] (Universidade do Porto, Portugal). Their results were later developed by Dmitry Korzun in his PhD [2] (PetrSU, under scientific supervising of Dr. Yury Bogoyavlenskiy) with introducing new efficient polynomial algorithms for solving some classes of ANLDE-systems [3]. These algorithms are called "syntactic" because of their grammar-based nature.

Nowadays Web is the standard way to introduce results to whom it may concern. This Project is intended to present the syntactic algorithms for comprehensive independent testing and to demonstrate for a wide spectrum of researches the computational efficiency of the algorithms. The target web system should visually, on manually and automatically generated examples, present principal ideas of the syntactic algorithms, demonstrate to a user what computational behavior they have, and how efficiently they work comparing with some available algorithms of other authors.

This Project is a student software engineering (SE) project in the framework of cooperation between the CS Departments of PetrSU and the University of Helsinki (UH). The developers are students of the CS Dept. of PetrSU. They study course "Software Engineering". The Project for them is an exercise for applying SE techniques in practice. The CS Dept. of UH helps the PetrSU students to study these SE topics, consults them on the related questions, and makes an expert estimation of the process.

The short name of the Project is **Web-SynDic**. This means that the target system is Web-based; "SynDic" is the word "Syntactic" (the algorithms are syntactic), where "tact" is replaced with "D" according to our tactful respect to Diophant as a father of NLDE Theory. Based on formal grammars theory, just use the rule $\boxed{\text{tact} \to \text{D}}$ to transform the string "Syntactic" to the string "SynDic".

2 Project Team

Modification History

Version	Date	Author(s)	Modification	Reason (optional)
1.00 (2.1)	16.07.2003	D. Korzun, Y. Bogoyavlenskiy	Original team	
1.10 (2.2)	24.08.2003	D. Korzun	Links to the personal web pages have been added.	More details about the team ought to be published.

2.1 Project Team: Version 1.00

2.1.1 Client part

Yury A. Bogoyavlenskiy (Head of the Department of Computer Science, University of Petrozavodsk, Ph.D., associate Professor). The Customer of the Project.
`Yury.Bogoyavlenskiy@cs.Karelia.RU`

Vadim A. Ponomarev (Lecturer, Department of Computer Science, University of Petrozavodsk, Ms.Sc., system administrator). The customer Expert and Consultant.

Vadim.Ponomarev@cs.Karelia.RU

2.1.2 Experts and Advisors

Timo Alanko (Vice-Head of the Department of Computer Science, University of Helsinki, Ph.D., Dr.hc., Professor).
Timo.Alanko@cs.Helsinki.FI

Inkeri Verkamo (Professor, Department of Computer Science, University of Helsinki, Ph.D.).
Inkeri.Verkamo@cs.Helsinki.FI

Juha Taina (Lecturer, Department of Computer Science, University of Helsinki, Ph.D.).
Juha.Taina@cs.Helsinki.FI

Turjo Tuohiniemi (Lecturer, Department of Computer Science, University of Helsinki).
Turjo.Tuohiniemi@cs.Helsinki.FI

2.1.3 Developers

Dmitry G. Korzun (Senior lecturer, Department of Computer Science, University of Petrozavodsk, Ph.D.). The Senior Manager and Instructor.
Dmitri.Korzun@cs.Karelia.RU

Kirill A. Kulakov (Master student, Department of Computer Science, University of Petrozavodsk). The Technical Manager.
Kirill.Kulakov@cs.Karelia.RU

Petr A. Semin (Master student, Department of Computer Science, University of Petrozavodsk). The Technical Manager; the Secretary (16.07–31.07.2003)
Petr.Syomin@cs.Karelia.RU

Andrey Y. Salo (senior student, Department of Computer Science, University of Petrozavodsk). The Senior Practitioner.
Andrei.Salo@cs.Karelia.RU

Andrey V. Anan'in (junior student, Department of Computer Science, University of Petrozavodsk). The Practitioner.
Andrei.Ananin@cs.Karelia.RU

Mikhail A. Kryshen' (junior student, Department of Computer Science, University of Petrozavodsk). The Practitioner.
Mihail.Kryshen@cs.Karelia.RU

Natalia I. Rodchenkova (Master student, Department of Mathematical Modelling and Control Systems, University of Petrozavodsk). The Secretary (August–November, 2003).
Natalya.Rodchenkova@cs.Karelia.RU

2.2 Project Team: Version 1.10

2.2.1 Client part

Yury A. Bogoyavlenskiy (Head of the Department of Computer Science, University of Petrozavodsk, Ph.D., associate Professor). The Customer of the Project.
Yury.Bogoyavlenskiy@cs.Karelia.RU

Vadim A. Ponomarev (Lecturer, Department of Computer Science, University of Petrozavodsk, Ms.Sc., system administrator). The customer Expert and Consultant.
Vadim.Ponomarev@cs.Karelia.RU

2.2.2 Experts and Advisors

Timo Alanko (Vice-Head of the Department of Computer Science, University of Helsinki, Ph.D., Dr.hc., Professor).
Timo.Alanko@cs.Helsinki.FI

Inkeri Verkamo (Professor, Department of Computer Science, University of Helsinki, Ph.D.).
Inkeri.Verkamo@cs.Helsinki.FI

Juha Taina (Lecturer, Department of Computer Science, University of Helsinki, Ph.D.).
Juha.Taina@cs.Helsinki.FI

Turjo Tuohiniemi (Lecturer, Department of Computer Science, University of Helsinki).
Turjo.Tuohiniemi@cs.Helsinki.FI

2.2.3 Developers

Dmitry G. Korzun (Senior lecturer, Department of Computer Science, University of Petrozavodsk, Ph.D.). The Senior Manager and Instructor.
Dmitri.Korzun@cs.Karelia.RU

Kirill A. Kulakov (Master student, Department of Computer Science, University of Petrozavodsk). The Technical Manager.
Kirill.Kulakov@cs.Karelia.RU

Petr A. Semin (Master student, Department of Computer Science, University of Petrozavodsk). The Technical Manager; the Secretary (16.07–31.07.2003)
`Petr.Syomin@cs.Karelia.RU`

Andrey Y. Salo (senior student, Department of Computer Science, University of Petrozavodsk). The Senior Practitioner.
`Andrei.Salo@cs.Karelia.RU`

Andrey V. Anan'in (junior student, Department of Computer Science, University of Petrozavodsk). The Practitioner.
`Andrei.Ananin@cs.Karelia.RU`

Mikhail A. Kryshen' (junior student, Department of Computer Science, University of Petrozavodsk). The Practitioner.
`Mihail.Kryshen@cs.Karelia.RU`

Natalia I. Rodchenkova (Master student, Department of Mathematical Modelling and Control Systems, University of Petrozavodsk). The Secretary (August–November, 2003).
`Natalya.Rodchenkova@cs.Karelia.RU`

3 Problem Domain

Modification History

Version	Date	Author(s)	Modification	Reason (optional)
1.00 (3.1)	??.07.2003	D. Korzun	under construction	

3.1 Problem Domain: Version 1.00

3.1.1 Nonnegative Linear Diophantine Equations

Let $\mathbb{Z} = \{0, \pm 1, \pm 2, \ldots\}$ be a set of integers, $\mathbb{Z}_+ = \{0, 1, 2, \ldots\}$ be a set of nonnegative integers.

A system of nonnegative linear Diophantine equations (NLDE for short) is a linear system

$$Ax = b, \qquad A \in \mathbb{Z}^{n \times m}, \quad b \in \mathbb{Z}^n, \quad x \in \mathbb{Z}_+^m, \tag{1}$$

where n is number of equations, m is number of unknowns. Therefore, the solutions of a NLDE system are m-vectors with nonnegative integer components.

For example, consider the NLDE system with $n = 2$ equations and $m = 4$ unknowns:

$$\begin{cases} x_1 - x_2 + 3x_3 = 1 \\ x_1 + 2x_2 - x_4 = -1 \end{cases} \tag{2}$$

This system has infinitely many solutions in $\mathbb{Z}_+^4$. Among them, for instance, solutions $x = (0, 2, 1, 5)^\top$ and $x' = (2, 4, 1, 11)^\top$.

NLDE system (1) is *homogenous*, if b is null vector $\mathbb{O} \in \mathbb{Z}^n$. Any homogenous NLDE system has the trivial solution $x = \mathbb{O} \in \mathbb{Z}_+^m$.

For any NLDE system a corresponding homogenous system can be constructed replacing right-hand side b by $\mathbb{O}$. For example, the corresponding homogenous NLDE system for (2) is

$$\begin{cases} x_1 - x_2 + 3x_3 = 0 \\ x_1 + 2x_2 - x_4 = 0 \end{cases} \tag{3}$$

If x is a solution of (1) and h is a solution of the corresponding homogeneous system, then $x' = x + \alpha h$ is also a solution of (1) for any nonnegative integer α.

Nontrivial solution h of homogenous NLDE system is *indecomposable* if it is not a sum of two nontrivial solutions of the same homogenous system. Solution x of NLDE system (1) is *indecomposable* if it is not a sum of another solution x' and a nontrivial solution h of the corresponding homogenous system. For instance, the solution $x' = (2, 4, 1, 11)^\top = (0, 2, 1, 5)^\top + (2, 2, 0, 6)^\top$ of NLDE system (2) is decomposable. The solution $h' = (2, 2, 0, 6)^\top = (1, 1, 0, 3)^\top + (1, 1, 0, 3)^\top$ of the homogenous system (3) is decomposable. The solutions $x = (0, 2, 1, 5)^\top$ and $h = (1, 1, 0, 3)^\top$ (of NLDE systems (2) and (3) respectively) are indecomposable.

The notion "a indecomposable solution" for NLDE systems is equivalent to "a minimal solution", where the minimality is considered with respect to standard component-wise ordering. For example, $x = (0, 2, 1, 5)^\top$ is minimal (for system (2)) and, therefore, there is no another solution x' such that $x' \leq x$; $h = (1, 1, 0, 3)^\top$ is minimal (for system (3)) and, therefore, there is no another (nontrivial) solution h' such that $h' \leq h$

Hilbert basis of NLDE system (1) is a pair $(\mathcal{N}, \mathcal{H})$, where $N \subset \mathbb{Z}_+^m$ is the set of all minimal (indecomposable) solutions of the NLDE system, and $\mathcal{H}$ is the set of all minimal (indecomposable) solutions of the corresponding homogenous system.

Sets $\mathcal{N}$ and $\mathcal{H}$ are always finite. They describe the set of all solutions of (1). Namely, any solution x can be computed as:

$$x = x^{(l)} + \sum_{s=1}^{q} \alpha_s h^{(s)} \quad \text{for some } x^{(l)} \in \mathcal{N} \text{ and } \alpha_s \in \mathbb{Z}_+, \tag{4}$$

where p and q are the numbers of solutions in the basis: $\mathcal{N} = \{x^{(1)}, \ldots, x^{(p)}\}$, $\mathcal{H} = \{h^{(1)}, \ldots, h^{(q)}\}$.

For example, all solutions of NLDE system (2) are described by the formula:

$$x = \left[\begin{array}{l} (1, 0, 0, 2)^\top + \alpha(1, 1, 0, 3)^\top + \beta(0, 3, 1, 6)^\top \\ (0, 2, 1, 5)^\top + \alpha(1, 1, 0, 3)^\top + \beta(0, 3, 1, 6)^\top \end{array} \right. \quad \text{for } \forall\, \alpha, \beta \in \mathbb{Z}_+,$$

where the left square bracket means alternation, $p = q = 2$, $\mathcal{N} = \{x^{(1)} = (1,0,0,2)^\top, x^{(2)} = (0,2,1,5)^\top\}$, $\mathcal{H} = \{h^{(1)} = (1,1,0,3)^\top, h^{(2)} = (0,3,1,6)^\top\}$. Taking $x^{(2)} = (0,2,1,5)^\top$, $\alpha = 2$, $\beta = 0$ we get the solution $x' = (2,4,1,11)^\top$.

This is very simular to the theory of linear equations: general solution is a sum of a particular solution and a linear combination of homogenous ones. However, the NLDE case has its own specifics: one particular solution is not enough to describe all solutions; there must be a set (but finite) of particular solutions. For the above example, at least two particular solutions are required.

See monograph [4] and papers [5, 6] as classic introduction to the area.

3.1.2 Complexity problems for NLDE systems

The basic computation problems for a given NLDE system are listed below.

1. Is the NLDE system solvable in nonnegative integers? (for the homogeneous case, only nontrivial solvability is considered.)

2. Searching a particular solution of the NLDE system (if any). It may include searching any particular solution or any minimal solution or a particular solution satisfied to some other criteria.

3. Searching Hilbert basis of the NLDE system.

The list can be prolonged. For instance, an interesting problem is counting basis solutions.

The listed problems are very complex in the computational sense. The solvability and particular solution searching problems are NP-complete [4]. The problem of searching Hilbert basis is even more complex—overNP.

This is a reason for dicovering particular classes of NLDE systems that have efficient (polynimial) algorithms for solving. We suggest (basing on the original work of M. Filgueiras and A.-P. Tomás [1]) an interesting approach for this—using formal grammars to establish such classes of NLDE systems.

A formal grammar is assigned to a NLDE system and solution searching is moved from set $\mathbb{Z}_+^m$ to a set of derivations in the grammar—one searches a derivation instead of a NLDE solution (see Figure 1). The idea is the same as in operations calculus, where the problem of solving an integral-differential equation is reduced to solving an algebraic equation in complex numbers.

The methods and algorithms of the formal grammars theory have been very developed starting from 50th. Our approach turns out that they can be fruitfully used to construct really efficient algorithms for solving some particular classes of NLDE system. The efficiency means

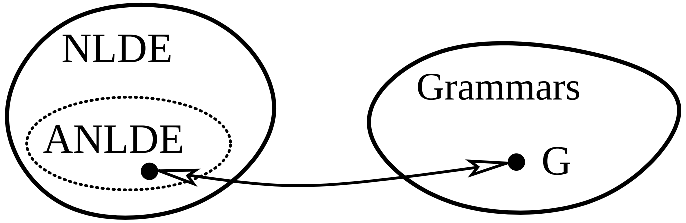

Figure 1: Association between NLDE and formal grammars: a general view

here that the algorithms are polynomial or pseudopolynomial (comparing with the general NP-complete and overNP case with exponential complexity). The pseudopolynomiality allows to select those problems (from the set of complex ones) that are permissible for practical solving.

3.1.3 NLDE systems, associated with CF-grammars. The syntactic algorithms

Consider a CF-grammar $G = (N, \Sigma, P, \cdot)$ and two strings v and w. Let G have m rules ($P = \{r_1, r_2, \ldots, r_m\}$), n nonterminals ($N = \{A_1, A_2, \ldots, A_n\}$), and t terminals ($\Sigma = \{a_1, a_2, \ldots, a_t\}$). Grammar rules r_i has the form $A \to \alpha$, where α is a string in $(N \cup \Sigma)^*$. The start nonterminal symbol is not required.

One can construct *an associated with the grammar NLDE system* (ANLDE system for short) as follows:

$$\begin{cases} \sum\limits_{i \in I_k} x_i - \sum\limits_{i=1}^{m} \gamma_{ki} x_i = b_k\ , \quad k = 1, 2, \ldots, n \ \text{ (i.e. for each nonterminal } A_k\text{)}, \\ \sum\limits_{i=1}^{m} \gamma_{ki} x_i = b_k\ , \quad k = n+1, n+2, \ldots, n+t \ \text{ (i.e. for each terminal } a_{k-n}\text{)}, \end{cases} \tag{5}$$

where $I_1 \cup \ldots \cup I_n = \{1, 2, \ldots, m\}$ is a partition of $\{1,\ldots,\mathrm{m}\}$, $\gamma_{ki} \in \mathbb{Z}_+$, $b_k \in \mathbb{Z}$. Let us denote this system as $S(G; v, w)$.

The ANLDE system has m unknowns (correspond to m grammar rules), $n + t$ equations (correspond to $n + t$ grammar symbols). Each set I_k contains indecies i such that rule r_i has right-hand side with nonternimal A_k, i.e. $A_k \to \alpha$. The coefficients γ_{ki} is the number of occurances of nonterminal A_k ($b_k = \mathrm{occ}(\alpha^{(i)}, A_k)$, $1 \le k \le n$) and terminal a_{k-n} ($b_k = \mathrm{occ}(\alpha^{(i)}, a_{k-n})$, $n + 1 \le k \le n + t$) in the right-hand side of the rule $r_i = A \to \alpha^{(i)}$. The coefficients b_k is the difference between the number of occurancies of A_k ($b_k = \mathrm{occ}(w, A_k) - \mathrm{occ}(v, A_k)$, $1 \le k \le n$) and a_{k-n} ($b_k = \mathrm{occ}(w, a_{k-n}) - \mathrm{occ}(v, a_{k-n})$, $n + 1 \le k \le n + t$) in the strings w and v.

For example, consider the grammar G_1 with nonterminals A and B ($n = 2$), the only terminal a ($t = 1$) and $m = 4$ rules (listed in Table 1, col. 1). Let $v = B$ and $w = aaAaaa$.

CF-grammar	ANLDE system (original)	ANLDE system (reduced)
$r_1: \ A \to AAB$ $r_2: \ A \to aBBa$ $r_3: \ B \to AAAaB$ $r_4: \ B \to \varepsilon$	$A: \ (x_1+x_2)-(2x_1+3x_3)=-1$ $B: \ (x_3+x_4)-(x_1+2x_2+x_3)=1$ $a: \ 2x_2+x_3=5$	$\begin{cases} x_1-x_2+3x_3=1 \\ x_1+2x_2-x_4=-1 \\ 2x_2+x_3=5 \end{cases}$

ε is emty string

Each grammar rule is marked with corresponding r_i

Each equation of the ANLDE system is market with a coresponding grammar symbol

Table 1: An example of ANLDE system

Then the ANLDE system for the grammar G_1 has the form, presented in Table 1, col. 2. The simplified system is shown in col. 3.

Let us note the requirement $I_1 \cup \ldots \cup I_n = \{1, 2, \ldots, m\}$ may be reduced to $I_1 \cup \ldots \cup I_n \subset \{1, 2, \ldots, m\}$. If there is some $i \notin I_1 \cup \ldots \cup I_n$, then one can add x_i to both sides of some equation (but exactly to one equation!). For example, the NLDE system

$$\begin{cases} (x_1+x_2)-(2x_1+3x_3)=-1 \\ x_4-(x_1+2x_2)=1 \\ 2x_2+x_3=5 \end{cases}$$

is not in form (5), because $3 \notin I_1 \cup I_2 = \{1, 2\} \cup \{4\}$. Adding x_3 to the second equation we get the same ANLDE system as in Table 1

The following theorem shows the relation between derivations in a CF-grammar and solutions of the ANLDE system.

Theorem 1. *Let $v \Rightarrow^* w$ be a derivation in CF-grammar G, x_i is the number of applications of rule r_i in this derivation (i=1,2,…,m). Then $x = (x_1, x_2, \ldots, x_m)$ is a solution of ANLDE system $S(G; v, w)$.*

Therefore, for the ANLDE system form Table 1, each succesfull derivation $B \Rightarrow^* aaAaaa$ corresponds to a solution. Consider such a derivation:

$$B \overset{r_3}{\Longrightarrow} AAAaB \overset{r_4}{\Longrightarrow} AAAa \overset{r_2}{\Longrightarrow} aBBaAAa \overset{r_4,r_4}{\Longrightarrow} aaAAa \overset{r_2}{\Longrightarrow} aaAaBBaa \overset{r_4,r_4}{\Longrightarrow} aaAaaa$$

Rule r_1 is not applied, so $x_1 = 0$; rule r_2 is applied two times, so $x_2 = 2$; rule r_3 is applied one time, $x_3 = 1$; rule r_4 is applied five times, $x_4 = 5$. Thus, we get a solution $x = (0, 2, 1, 5)^\top$.

Definition of ANLDE system does not take into account the order of symbols in strings (v, w, right-hand sides of rules r_i). Let us define $\{\alpha\}^\circ$ as a multiset of all symbols of α (the number of occurances is taken into account!). For example, $\{abaaabA\}^\circ = \{A, a, a, a, a, b, b\} = \{A^1, a^4, b^2\}$. Using this notation, $\{\alpha\}^\circ = \{\beta\}^\circ$ means that the strings α and β differ in order of symbols only. Binary operation $\triangle$ is a symmetric difference of two multisets.

Theorem 2. *Let G' and G'' are CF-grammars with the same alphabet $N \cup \Sigma$; v', w', v'', w'' are strings in this alphabet. If $\{\alpha'_i\}^\circ = \{\alpha''_i\}^\circ$ $\forall i = 1, 2, \ldots, m$, where α_i is a right-hand side of rule r_i, and $\{v'\}^\circ \triangle \{v''\}^\circ = \{w'\}^\circ \triangle \{w''\}^\circ$. Then $S(G'; v', w') = S(G''; v'', w'')$.*

According to the theorem, one can take the grammar G_1 (see the above example) and $(v, w) = (BA, aaAaaaA)$. ANLDE system are the same. The derivation

$$BA \overset{r_3}{\Longrightarrow} AAAaBA \overset{r_4}{\Longrightarrow} AAAaA \overset{r_2}{\Longrightarrow} aBBaAAaA \overset{r_4,r_4}{\Longrightarrow} aaAAaA \overset{r_2}{\Longrightarrow} aaAaBBaaA \overset{r_4,r_4}{\Longrightarrow} aaAaaaA$$

constructs the mentioned above solution $x = (0, 2, 1, 5)^\top$. However, for $(v, w) = (B, aaaaaA)$ there is no derivation $B \Rightarrow^* aaaaaA$ in the grammar G_1.

This means one does not take into accont i) the order of symbols in r_i, v, w and ii) symmetric addition/subtraction of symbols in v, w.

Theorem 3. *Vector $x \in \mathbb{Z}_+^m$ is a solution of ANLDE system $S(G, v, w)$ iff x corresponds to a derivation $v' \Rightarrow^* w'$, where $\{v\}^\circ \triangle \{v'\}^\circ = \{w\}^\circ \triangle \{w'\}^\circ$ and the order of symbols in sentential forms is ignored.*

3.1.4 Experimental analysis of the syntactic algorithms

3.1.5 Generating test ANLDE systems

3.1.6 Demonstrating and testing the syntactic algorithms

4 User Requirements

Modification History

Version	Date	Author(s)	Modification	Reason (optional)
1.00 (4.1)	16.07.2003	Y. Bogoyavlenskiy, D. Korzun	Initial draft	
1.01 (4.2)	18.07.2003	K. Kulakov	a minor textual error for Requirement AS4.	Discovered on a meeting
1.02 (4.3)	24.07.2003	D. Korzun	wrong sequence numbers for Requirements on security, see 4.3.3.	Discovered on a meeting
1.10 (4.4)	08.08.2003	Y. Bogoyavlenskiy	Customer's corrections.	Customer's inspection
1.20 (4.5)	17.08.2003	D. Korzun	Elicitation and detailing.	Inspection

4.1 User Requirements: Version 1.00

The aim of the Project is development of a web system for demonstrating the work of the syntactic algorithms for solving ANLDE systems. The users of the web system are researches that have an access to the Internet and use standard Internet browsers. The main objectives of the web system are the following.

- The key facts of the ANLDE systems theory are free available for the international scientific community.
- Visual tool for demonstrating and testing the efficiency of the syntactic algorithms on various examples including user's ones.
- Experience for the students what is the competent development of real software systems and how such a development is going on.

4.1.1 Development process

G1. The web system must be developed according with international principles, practice, standards and recommendations of SE. Use [7, 8, 9, 10] as basic books.

G2. The Project starts on 16.07.2003 and has to be completed in November, 2003.

G3. Working language of the Project is English. All artifacts are replicated in Russian.

G4. From the start of the project, students have to work 20 hours/week.

4.1.2 Functions of the web system

F1a. The web system must solve an input ANLDE system and show to a user the outcome. The input ANLDE system is given by a user or generated by the software (user choice).

F1b. In the case of Requirement F1a the outcome must include:

1. ANLDE system.
2. Corresponding CF-grammar (optional).
3. Solutions of the ANLDE system, e.g. Hilbert basis or a particular solution.
4. Derivations in the grammar corresponding to the solution (optional).
5. Metrics of the resource consumption by the syntactic algorithm (time and memory usage);

6. Comparative metrics for the alternative algorithms. (slopes, lp_solver, BonsaiG, GLPK).
7. Key hardware characteristics of the server.

F2a. The web system must solve a set of ANLDE systems. The input set is given by a user (TXT file) or generated by the software (user choice).

F2b. In the case of Requirement F2a the outcome must include:

1. Characteristics of the input set of ANLDE systems.
2. Statistics of the resource consumption by the syntactic algorithm (time and memory usage).
3. Comparative statistics for the alternative algorithms (see references for these algorithms in Requirement F1b).
4. Key hardware characteristics of the server.

F3. The web system must allow a user to send her/his opinion of the solution result. A special case here is user's disagreement with the found solution(s) of her/his ANLDE system (testing of the syntactic algorithms).

F4. The web system must register a user when she/he wishes. A registered user has a unique identifier (nick name).

F5. The web system must compute activity statistics of the registered users including time and resource consumption.

F6. The web system must compute activity statistics of all users. These users are identified by their IP-addresses.

4.1.3 Attributes of the web system

Usability

AU1. The traditional mathematical style must be supported for representation of ANLDE (and possibly NLDE) systems and their solutions.

AU2. The output must be available to a user in HTML and TXT formats.

AU3. Standard Internet browsers (Netscape, Mozilla, MS-IExplorer) must be supported.

Security

AS1. The demonstrated algorithms (ANLDE system solvers and test generators) must not be accessible to the external side. Only the outcomes of their work are available.

AS2. There are two types of users: regular ones and administrator.

AS3. The activity statistics must not be accessible for regular users.

AS3. For any regular user the web system must support default limits on the solution process (maximum time, memory, absolute values of coefficients, maximum number of equations, unknowns, ANLDE systems in a test set, solutions in Hilbert basis, etc.).

AS4. A regular user may manage her/his own limits on the solution process; these limits must not exceed the default ones (see Requirement A3).

Performance

AP1. The web system must serve concurrently up to 5 users (separate user sessions) without significant reduction of the performance.

AP2. The web system must not overload a base web server more than 75% of the total server workload.

AP3. The web system must reply on a user action less than after 20 seconds. The reply is either the required data, or a notification on the progress.

Deployment

AD1. Client part of the web system must be available to a user via an Internet browser without an explicit installation.

4.2 User Requirements: Version 1.01

The aim of the Project is development of a web system for demonstrating the work of the syntactic algorithms for solving ANLDE systems. The users of the web system are researches that have an access to the Internet and use standard Internet browsers. The main objectives of the web system are the following.

- The key facts of the ANLDE systems theory are free available for the international scientific community.

- Visual tool for demonstrating and testing the efficiency of the syntactic algorithms on various examples including user's ones.
- Experience for the students what is the competent development of real software systems and how such a development is going on.

4.2.1 Development process

G1. The web system must be developed according with international principles, practice, standards and recommendations of SE. Use [7, 8, 9, 10] as basic books.

G2. The Project starts on 16.07.2003 and has to be completed in November, 2003.

G3. Working language of the Project is English. All artifacts are replicated in Russian.

G4. From the start of the project, students have to work 20 hours/week.

4.2.2 Functions of the web system

F1a. The web system must solve an input ANLDE system and show to a user the outcome. The input ANLDE system is given by a user or generated by the software (user choice).

F1b. In the case of Requirement F1a the outcome must include:

1. ANLDE system.
2. Corresponding CF-grammar (optional).
3. Solutions of the ANLDE system, e.g. Hilbert basis or a particular solution.
4. Derivations in the grammar corresponding to the solution (optional).
5. Metrics of the resource consumption by the syntactic algorithm (time and memory usage);
6. Comparative metrics for the alternative algorithms.
(slopes, lp_solver, BonsaiG, GLPK).
7. Key hardware characteristics of the server.

F2a. The web system must solve a set of ANLDE systems. The input set is given by a user (TXT file) or generated by the software (user choice).

F2b. In the case of Requirement F2a the outcome must include:

1. Characteristics of the input set of ANLDE systems.

2. Statistics of the resource consumption by the syntactic algorithm (time and memory usage).
3. Comparative statistics for the alternative algorithms (see references for these algorithms in Requirement F1b).
4. Key hardware characteristics of the server.

F3. The web system must allow a user to send her/his opinion of the solution result. A special case here is user's disagreement with the found solution(s) of her/his ANLDE system (testing of the syntactic algorithms).

F4. The web system must register a user when she/he wishes. A registered user has a unique identifier (nick name).

F5. The web system must compute activity statistics of the registered users including time and resource consumption.

F6. The web system must compute activity statistics of all users. These users are identified by their IP-addresses.

4.2.3 Attributes of the web system

Usability

AU1. The traditional mathematical style must be supported for representation of ANLDE (and possibly NLDE) systems and their solutions.

AU2. The output must be available to a user in HTML and TXT formats.

AU3. Standard Internet browsers (Netscape, Mozilla, MS-IExplorer) must be supported.

Security

AS1. The demonstrated algorithms (ANLDE system solvers and test generators) must not be accessible to the external side. Only the outcomes of their work are available.

AS2. There are two types of users: regular ones and administrator.

AS3. The activity statistics must not be accessible for regular users.

AS3. For any regular user the web system must support default limits on the solution process (maximum time, memory, absolute values of coefficients, maximum number of equations, unknowns, ANLDE systems in a test set, solutions in Hilbert basis, etc.).

AS4. A regular user may manage her/his own limits on the solution process; these limits must not exceed the default ones (see Requirement AS3).

Performance

AP1. The web system must serve concurrently up to 5 users (separate user sessions) without significant reduction of the performance.

AP2. The web system must not overload a base web server more than 75% of the total server workload.

AP3. The web system must reply on a user action less than after 20 seconds. The reply is either the required data, or a notification on the progress.

Deployment

AD1. Client part of the web system must be available to a user via an Internet browser without an explicit installation.

4.3 User Requirements: Version 1.02

The aim of the Project is development of a web system for demonstrating the work of the syntactic algorithms for solving ANLDE systems. The users of the web system are researches that have an access to the Internet and use standard Internet browsers. The main objectives of the web system are the following.

- The key facts of the ANLDE systems theory are free available for the international scientific community.
- Visual tool for demonstrating and testing the efficiency of the syntactic algorithms on various examples including user's ones.
- Experience for the students what is the competent development of real software systems and how such a development is going on.

4.3.1 Development process

G1. The web system must be developed according with international principles, practice, standards and recommendations of SE. Use [7, 8, 9, 10] as basic books.

G2. The Project starts on 16.07.2003 and has to be completed in November, 2003.

G3. Working language of the Project is English. All artifacts are replicated in Russian.

G4. From the start of the project, students have to work 20 hours/week.

4.3.2 Functions of the web system

F1a. The web system must solve an input ANLDE system and show to a user the outcome. The input ANLDE system is given by a user or generated by the software (user choice).

F1b. In the case of Requirement F1a the outcome must include:

1. ANLDE system.
2. Corresponding CF-grammar (optional).
3. Solutions of the ANLDE system, e.g. Hilbert basis or a particular solution.
4. Derivations in the grammar corresponding to the solution (optional).
5. Metrics of the resource consumption by the syntactic algorithm (time and memory usage);
6. Comparative metrics for the alternative algorithms.
   (slopes, lp_solver, BonsaiG, GLPK).
7. Key hardware characteristics of the server.

F2a. The web system must solve a set of ANLDE systems. The input set is given by a user (TXT file) or generated by the software (user choice).

F2b. In the case of Requirement F2a the outcome must include:

1. Characteristics of the input set of ANLDE systems.
2. Statistics of the resource consumption by the syntactic algorithm (time and memory usage).
3. Comparative statistics for the alternative algorithms (see references for these algorithms in Requirement F1b).
4. Key hardware characteristics of the server.

F3. The web system must allow a user to send her/his opinion of the solution result. A special case here is user's disagreement with the found solution(s) of her/his ANLDE system (testing of the syntactic algorithms).

F4. The web system must register a user when she/he wishes. A registered user has a unique identifier (nick name).

F5. The web system must compute activity statistics of the registered users including time and resource consumption.

F6. The web system must compute activity statistics of all users. These users are identified by their IP-addresses.

4.3.3 Attributes of the web system

Usability

AU1. The traditional mathematical style must be supported for representation of ANLDE (and possibly NLDE) systems and their solutions.

AU2. The output must be available to a user in HTML and TXT formats.

AU3. Standard Internet browsers (Netscape, Mozilla, MS-IExplorer) must be supported.

Security

AS1. The demonstrated algorithms (ANLDE system solvers and test generators) must not be accessible to the external side. Only the outcomes of their work are available.

AS2. There are two types of users: regular ones and administrator.

AS3. The activity statistics must not be accessible for regular users.

AS4. For any regular user the web system must support default limits on the solution process (maximum time, memory, absolute values of coefficients, maximum number of equations, unknowns, ANLDE systems in a test set, solutions in Hilbert basis, etc.).

AS5. A regular user may manage her/his own limits on the solution process; these limits must not exceed the default ones (see Requirement AS3).

Performance

AP1. The web system must serve concurrently up to 5 users (separate user sessions) without significant reduction of the performance.

AP2. The web system must not overload a base web server more than 75% of the total server workload.

AP3. The web system must reply on a user action less than after 20 seconds. The reply is either the required data, or a notification on the progress.

Deployment

AD1. Client part of the web system must be available to a user via an Internet browser without an explicit installation.

4.4 User Requirements: Version 1.10

The aim of the Project is development of a web system for demonstrating the work of the syntactic algorithms for solving ANLDE systems. The users of the web system are researches that have an access to the Internet and use standard Internet browsers. The main objectives of the web system are the following.

- Presenting the key facts of the ANLDE systems theory for the international scientific community.
- Demonstrating and testing the efficiency of the syntactic algorithms with the visual tool on various examples including user's ones.
- Providing the student team with experience what is the process of competent development of real software systems.

4.4.1 Development process

G1. The web system must be developed according with international principles, practice, standards and recommendations of SE. Use [7, 8, 9, 10] as basic books.

G2. The Project starts on 16.07.2003 and has to be completed in November, 2003.

G3. Working language of the Project is English. All artifacts are replicated in Russian.

G4. From the start of the project, students have to work 20 hours/week.

4.4.2 Functions of the web system

F1a. The web system must solve an input ANLDE system and show to a user the outcome. The input ANLDE system is given by a user or generated by the software (user choice).

F1b. For the case of Requirement F1a the outcome must include:

1. ANLDE system.
2. Corresponding CF-grammar (optional).
3. Solutions of the ANLDE system, e.g. Hilbert basis or a particular solution.
4. Derivations in the grammar corresponding to the solution (optional).
5. Metrics of the resource consumption by the syntactic algorithm (time and memory usage);
6. Comparative metrics for the alternative algorithms.
(slopes, lp_solver, BonsaiG, GLPK).

7. Key hardware characteristics of the server.

F2a. The web system must solve a set of ANLDE systems. The input set is given by a user (TXT file) or generated by the software (user choice).

F2b. For the case of Requirement F2a the outcome must include:

1. Characteristics of the input set of ANLDE systems.
2. Statistics of the resource consumption by the syntactic algorithm (time and memory usage).
3. Comparative statistics for the alternative algorithms (see references for these algorithms in Requirement F1b).
4. Key hardware characteristics of the server.

F3. The web system must allow a user to send her/his opinion on the solution result. A special case here is user's disagreement with the found solution(s) of her/his ANLDE system (testing of the syntactic algorithms).

F4. The web system must register a user when she/he wishes. A registered user has a unique identifier (nick name).

F5. The web system must compute activity statistics of the registered users including time and resource consumption.

F6. The web system must compute activity statistics of all users. These users are identified by their IP-addresses.

4.4.3 Attributes of the web system

Usability

AU1. The traditional mathematical style must be supported for representation of ANLDE (and possibly NLDE) systems and their solutions.

AU2. The output must be available to a user in HTML and TXT formats.

AU3. Standard Internet browsers (Netscape, Mozilla, MS-IExplorer) must be supported.

Security

AS1. The demonstrated algorithms (ANLDE system solvers and test generators) must not be accessible to the external side. Only the outcomes of their work are available.

AS2. There are two types of users: regular ones and administrator.

AS3. The activity statistics must not be accessible for regular users.

AS4. For any regular user the web system must support default limits on the solution process (maximum time, memory, absolute values of coefficients, maximum number of equations, unknowns, ANLDE systems in a test set, solutions in Hilbert basis, etc.).

AS5. A regular user may manage her/his own limits on the solution process; these limits must not exceed the default ones (see Requirement AS3).

Performance

AP1. The web system must serve concurrently up to 5 users (separate user sessions) without significant reduction of the performance.

AP2. The web system must not overload a base web server more than 75% of the total server workload.

AP3. The web system must reply on a user action less than after 20 seconds. The reply is either the required data, or a notification on the progress.

Deployment

AD1. Client part of the web system must be available to a user via an Internet browser without an explicit installation.

4.5 User Requirements: Version 1.20

The aim of the Project is development of a web system for demonstrating the work of the syntactic algorithms for solving ANLDE systems. The users of the web system are researches that have an access to the Internet and use standard Internet browsers. The main objectives of the web system are the following.

- Presenting the key facts of the ANLDE systems theory for the international scientific community.

- Demonstrating and testing the efficiency of the syntactic algorithms with the visual tool on various examples including user's ones.
- Providing the student team with the experience what the process of competent development of real software systems is.

4.5.1 Development process

G1. The web system must be developed according with international principles, practice, standards and recommendations of SE. Use [7, 8, 9, 10] as basic books.

G2. The Project starts on 16.07.2003 and has to be completed in November, 2003.

G3. Working language of the Project is English. All artifacts are replicated in Russian.

G4. From the start of the project, students have to work 20 hours/week.

4.5.2 Functions of the web system

F1a. The web system must solve an input ANLDE system (homogeneous) and show to a user the outcome. The input ANLDE system is given by a user or generated by the software (user choice). These input systems are considered as test ANLDE systems. Only homogeneous ANLDE are used.

F1b. For the case of Requirement F1a the outcome must include:

1. Test ANLDE system.
2. Solutions of the ANLDE system, e.g. Hilbert basis or a particular solution.
3. Metrics of the resource consumption by the syntactic algorithm (time and memory usage estimates);
4. Comparative metrics for the alternative algorithms.
   (slopes, lp_solver, BonsaiG, GLPK).
5. Key hardware characteristics of the server.

F2a. The web system must solve a set of test ANLDE systems (homogeneous). The input test set is given by a user (TXT file) or generated by the software (user choice). Only homogeneous ANLDE are used.

F2b. For the case of Requirement F2a the outcome must include:

1. Characteristics of the input set of ANLDE systems.

2. Statistics of the resource consumption by the syntactic algorithm (time and memory usage estimates).
3. Comparative statistics for the alternative algorithms (see references for these algorithms in Requirement F1b).
4. Key hardware characteristics of the server.

F3. The web system must allow a user to send her/his opinion on the solution result. A special case here is user's explicit disagreement with the found solution(s) of the processed test ANLDE system (testing of the syntactic algorithms). The test ANLDE system may be included to the opinion.

F4. The web system must register a user when she/he wishes. A registered user has a unique identifier (nick name).

F5. The web system must compute activity statistics of the registered users including time and resource consumption (available for system administrator only).

F6. The web system must compute activity statistics of all users. These users are identified by their IP-addresses (available for system administrator only).

4.5.3 Attributes of the web system

Usability

AU1. The traditional mathematical style must be supported for representation of ANLDE (and possibly NLDE) systems and their solutions.

AU2. The output must be available to a user in HTML and TXT formats.

AU3. Standard Internet browsers (among them Netscape, Mozilla, MS-IExplorer) must be supported.

Security

AS1. The demonstrated algorithms (ANLDE system solvers and test generators) must not be accessible to the external side. Only the outcomes of their work are available.

AS2. There are two types of users: regular ones and a system administrator.

AS3. The activity statistics must not be accessible for regular users.

AS4. For any regular user the web system must support default limits on the solution process (maximum time, memory, absolute values of coefficients, maximum number of equations, unknowns, ANLDE systems in a test set, solutions in Hilbert basis, etc.).

AS5. A regular user may manage her/his own limits on the solution process; these limits must not exceed the default ones (see Req. AS4).

Performance

AP1. The web system must serve concurrently up to 5 users (separate user sessions) without significant reduction of the performance.

AP2. The web system must not overload a base web server more than 75% of the total server workload.

AP3. The web system must reply on a user action less than after 20 seconds. The reply is either the required data, or a notification on the progress.

Deployment

AD1. Client part of the web system must be available to a user via an Internet browser without an explicit installation.

5 Project Plan

Modification History

Version	Date	Author(s)	Modification	Reason (optional)
1.00 (5.1)	16.07.2003	D. Korzun	Preliminary draft	
1.01 (5.2)	11.08.2003	D. Korzun	Requirements analysis phase is prolonged.	It takes more time than we planned.
1.02 (5.3)	17.08.2003	D. Korzun	Adding details, modifying dates.	Senior Manager's inspection.
1.10 (5.4)	20.09.2003	D. Korzun	Design phase is updated.	Senior Manager's inspection.
1.11 (5.5)	01.10.2003	D. Korzun	milestones are updated—the project is delayed again.	Senior Manager's inspection.

1.12 (5.6)	20.10.2003	D. Korzun	milestones are updated—the project is delayed again.	Senior Manager's inspection.
1.13 (5.7)	28.10.2003	D. Korzun	Changes according with the implementation phase	Senior Manager's inspection.
1.20 (5.8)	21.11.2003	D. Korzun	Implementation phase is split into unit implementation and integration	Senior Manager's offer.
1.21 (5.9)	28.11.2003	D. Korzun	Unit implementation is delayed.	Senior Manager's offer.
1.22 (5.10)	13.12.2003	D. Korzun	Integration is delayed	Senior Manager's offer.
2.00 (5.11)	10.01.2004	D. Korzun	System testing (includes alpha-testing, publishing the system, and beta-testing.	Working product is completed, but not a stable version.
2.10 (5.12)	17.02.2004	D. Korzun	The 2nd stage of alpha-testing is moved for later time	Not enough human resources.

5.1 Project Plan: Version 1.00

The waterfall model is chosen for the software process. The Customer may slightly change the requirements during "Requirements analysis" and "Design" stages; after that the requirements are frozen.

The start point of the Project is 7.07.2003; at this date the planning phase is initiated. The real development process starts 16.07.2003 when the first meeting of the project team is conducted. A working version of the required web system must be completed as the conclusion of the Project (November, 2003).

5.1.1 Stages

0. Initial planning (7.07–16.07.2003). Definition of the Project objectives and scope. Customer and Senior Manager discussing. Enrolling student software team. The skeleton of the maintenance document is produced (this one).

The main artifacts:

- Version 1.00 of the User Requirements (see section 4),
- Version 1.00 of the project plan (this section).

1. Requirements analysis (4 weeks, 16.07–12.08.2003). Analysis of the problem domain and user requirements.

The main artifacts: The requirement specification document.

- Glossary of the problem domain.
- Requirements elicitation, simplification and priority.
- Requirements detailing.
- Use case analysis.
- Conceptual model.
- Requirements validation.

2. Design(4 weeks, 13.08–9.09.2003).

3. Implementation (4 weeks, 10.08–7.10.2003).

4. Testing (4 weeks, 8.10–4.11.2003).

Conclusion of the Project (November, 2003).

5.2 Project Plan: Version 1.01

The waterfall model is chosen for the software process. The Customer may slightly change the requirements during "Requirements analysis" and "Design" stages; after that the requirements are frozen.

The start point of the Project is 7.07.2003; at this date the planning phase is initiated. The real development process starts 16.07.2003 when the first meeting of the project team is conducted. A working version of the required web system must be completed as the conclusion of the Project (November, 2003).

5.2.1 Stages

0. Initial planning (7.07–16.07.2003). Definition of the Project objectives and scope. Customer and Senior Manager discussing. Enrolling student software team. The skeleton of the maintenance document is produced (this one).

The main artifacts:

- Version 1.00 of the User Requirements (see section 4),
- Version 1.00 of the project plan (section 5.1).

1. **Requirements analysis** (16.07–25.08.2003). Analysis of the problem domain and user requirements.
   The main artifacts: The Requirement specification document.

   - Expanded User Requirements (13.08.2003)
   - Glossary of the problem domain (15.08.2003).
   - Conceptual model of the Problem domain (18.08.2003).
   - High-level architecture (18.03.2003)
   - High-level use cases, use case diagram (15.03.2003)
   - System requirements (22.03.2003)
   - Expanded use cases, sequence diagrams (22.03.2003)
   - Requirements validation (25.08.2003).

2. **Design**(26.08–30.09.2003).

3. **Implementation** (1.10–24.10.2003).

4. **Testing** (25.10–18.11.2003).

Conclusion of the Project (November, 2003).

5.3 Project Plan: Version 1.02

The waterfall model is chosen for the software process. The Customer may slightly change the requirements during “Requirements analysis” and “Design” phases; after that the requirements are frozen.

The start point of the Project is 7.07.2003; at this date the planning phase is initiated. The real development process starts 16.07.2003 when the first meeting of the project team is conducted. A working version of the required web system must be completed as the conclusion of the Project (November, 2003).

5.3.1 Phases

0. **Initial planning** (7.07–16.07.2003). Definition of the Project objectives and scope. Customer and Senior Manager discussing. Enrolling student software team. The skeleton of the maintenance document is produced (this one).
   The main artifacts:

   - Version 1.00 of the User Requirements (sect. 4.1),

   - Version 1.00 of the Project Plan (sect. 5.1).

1. **Requirements analysis** (16.07–25.08.2003). Analysis of the problem domain and user requirements.
   The main artifacts: Requirements Specification:

   - Expanded User Requirements (13-15.08.2003, inspection 20.08)
   - Glossary of the problem domain (15.08.2003, inspection 20.08).
   - Conceptual model of the Problem domain (18.08.2003, inspection 20.08).
   - High-level architecture (18.03.2003, inspections 20-22.08)
   - High-level use cases, use case diagram (18.03.2003, inspection 20.08)
   - System requirements (22.03.2003, inspections 22-25.08)
   - Expanded use cases, sequence diagrams (22.03.2003, inspection 22-25.08)
   - Requirements validation (25.08.2003).

   See also the later versions of General (sect. 1), User Requirements (sect. 4), and Project Plan (sect. 5).

2. **Design** (26.08–30.09.2003). User interface design, Architecture design, External algorithms interface design, Test cases design.

3. **Implementation** (1.10–24.10.2003).

4. **Testing** (25.10–18.11.2003).

Conclusion of the Project (November, 2003).

5.4 Project Plan: Version 1.10

The waterfall model is chosen for the software process. The Customer may slightly change the requirements during “Requirements analysis” and “Design” phases; after that the requirements are frozen.

The start point of the Project is 7.07.2003; at this date the planning phase is initiated. The real development process starts 16.07.2003 when the first meeting of the project team is conducted. A working version of the required web system must be completed as the conclusion of the Project (November, 2003).

5.4.1 Phases

0. Initial planning (7.07–16.07.2003). Definition of the Project objectives and scope. Customer and Senior Manager discussing. Enrolling student software team. The skeleton of the maintenance document is produced (this one).
The main artifacts:

- Version 1.00 of the User Requirements (sect. 4.1),
- Version 1.00 of the Project Plan (sect. 5.1).

1. Requirements analysis (16.07–25.08.2003). Analysis of the problem domain and user requirements.
The main artifacts: Requirements Specification:

- Expanded User Requirements (13-15.08.2003, inspection 20.08)
- Glossary of the problem domain (15.08.2003, inspection 20.08).
- Conceptual model of the Problem domain (18.08.2003, inspection 20.08).
- High-level architecture (18.03.2003, inspections 20-22.08)
- High-level use cases, use case diagram (18.03.2003, inspection 20.08)
- System requirements (22.03.2003, inspections 22-25.08)
- Expanded use cases, sequence diagrams (22.03.2003, inspection 22-25.08)
- Requirements validation (25.08.2003).

See also the later versions of General (sect. 1), User Requirements (sect. 4), and Project Plan (sect. 5).

2. Design (26.08–10.10.2003). The main artifacts: Design Specification:

- Architecture design (22.09.2003, analysis & inspection 22-25.09).
- User interface design (22.09.2003, analysis & inspection 22-25.09).
- Subsystems interface design (25.09.2003, analysis & inspection 25-27.09).
- Subsystems design (1.10.2003, analysis & inspection 2-4.10).
- Test plan & test cases design (4.10.2003, analysis & inspection 5-8.10).
- Design Specification (10.10.2003).

See also the later versions of User Requirements (sect. 4), and Project Plan (sect. 5).

3. Implementation (10.10–6.11.2003).

4. Testing (8.11–28.11.2003).

Conclusion of the Project (November/December, 2003).

5.5 Project Plan: Version 1.11

The waterfall model is chosen for the software process. The Customer may slightly change the requirements during “Requirements analysis” and “Design” phases; after that the requirements are frozen.

The start point of the Project is 7.07.2003; at this date the planning phase is initiated. The real development process starts 16.07.2003 when the first meeting of the project team is conducted. A working version of the required web system must be completed as the conclusion of the Project (November, 2003).

5.5.1 Phases

0. Initial planning (7.07–16.07.2003). Definition of the Project objectives and scope. Customer and Senior Manager discussing. Enrolling student software team. The skeleton of the maintenance document is produced (this one).
The main artifacts:

- Version 1.00 of the User Requirements (sect. 4.1),
- Version 1.00 of the Project Plan (sect. 5.1).

1. Requirements analysis (16.07–25.08.2003). Analysis of the problem domain and user requirements.
The main artifacts: Requirements Specification:

- Expanded User Requirements (13-15.08.2003, inspection 20.08)
- Glossary of the problem domain (15.08.2003, inspection 20.08).
- Conceptual model of the Problem domain (18.08.2003, inspection 20.08).
- High-level architecture (18.03.2003, inspections 20-22.08)
- High-level use cases, use case diagram (18.03.2003, inspection 20.08)
- System requirements (22.03.2003, inspections 22-25.08)
- Expanded use cases, sequence diagrams (22.03.2003, inspection 22-25.08)
- Requirements validation (25.08.2003).

See also the later versions of General (sect. 1), User Requirements (sect. 4), and Project Plan (sect. 5).

2. Design (26.08–10.10.2003). The main artifacts: Design Specification:

- Architecture design (last draft 6.10.2003, final version 8.10).
- User interface design (last draft 6.10.2003, final version 10.10).
- Subsystems interface design (16.10.2003, analysis & inspection 16-18.10).
- Subsystems design (17.10.2003, analysis & inspection 17-20.10).
- Test plan & test cases design (20.10.2003, analysis & inspection 20-23.10).
- Design Specification (22.10.2003).

3. Implementation (27.10–16.11.2003).

4. Testing (17.11–28.11.2003).

Conclusion of the Project (November/December, 2003).

5.6 Project Plan: Version 1.12

The waterfall model is chosen for the software process. The Customer may slightly change the requirements during "Requirements analysis" and "Design" phases; after that the requirements are frozen.

The start point of the Project is 7.07.2003; at this date the planning phase is initiated. The real development process starts 16.07.2003 when the first meeting of the project team is conducted. A working version of the required web system must be completed as the conclusion of the Project (November, 2003).

5.6.1 Phases

0. Initial planning (7.07–16.07.2003). Definition of the Project objectives and scope. Customer and Senior Manager discussing. Enrolling student software team. The skeleton of the maintenance document is produced (this one).
The main artifacts:

- Version 1.00 of the User Requirements (sect. 4.1),
- Version 1.00 of the Project Plan (sect. 5.1).

1. Requirements analysis (16.07–25.08.2003). Analysis of the problem domain and user requirements.
The main artifacts: Requirements Specification:

- Expanded User Requirements (13-15.08.2003, inspection 20.08)

- Glossary of the problem domain (15.08.2003, inspection 20.08).
- Conceptual model of the Problem domain (18.08.2003, inspection 20.08).
- High-level architecture (18.03.2003, inspections 20-22.08)
- High-level use cases, use case diagram (18.03.2003, inspection 20.08)
- System requirements (22.03.2003, inspections 22-25.08)
- Expanded use cases, sequence diagrams (22.03.2003, inspection 22-25.08)
- Requirements validation (25.08.2003).

See also the later versions of General (sect. 1), User Requirements (sect. 4), and Project Plan (sect. 5).

2. Design (26.08–27.10.2003). The main artifacts: Design Specification:

- Architecture design (last draft 6.10.2003, final version 8.10).
- User interface design (last draft 6.10.2003, final version 10.10).
- Subsystems design (25.10.2003, analysis & inspection 25-27.10).
- Test plan & test cases design (29.10.2003, analysis & inspection 29-31.10).

3. Implementation (27.10–16.11.2003). including unit testing.

4. Testing (17.11–28.11.2003). Integration and validation testing.

Conclusion of the Project (December, 2003).

5.7 Project Plan: Version 1.13

The waterfall model is chosen for the software process. The Customer may slightly change the requirements during "Requirements analysis" and "Design" phases; after that the requirements are frozen.

The start point of the Project is 7.07.2003; at this date the planning phase is initiated. The real development process starts 16.07.2003 when the first meeting of the project team is conducted. A working version of the required web system must be completed as the conclusion of the Project (November, 2003).

5.7.1 Phases

0. Initial planning (7.07–16.07.2003). Definition of the Project objectives and scope. Customer and Senior Manager discussing. Enrolling student software team. The skeleton of the maintenance document is produced (this one).
The main artifacts:

- Version 1.00 of the User Requirements (sect. 4.1),
- Version 1.00 of the Project Plan (sect. 5.1).

1. Requirements analysis (16.07–25.08.2003). Analysis of the problem domain and user requirements.
The main artifacts: Requirements Specification:

- Expanded User Requirements (13-15.08.2003, inspection 20.08)
- Glossary of the problem domain (15.08.2003, inspection 20.08).
- Conceptual model of the Problem domain (18.08.2003, inspection 20.08).
- High-level architecture (18.03.2003, inspections 20-22.08)
- High-level use cases, use case diagram (18.03.2003, inspection 20.08)
- System requirements (22.03.2003, inspections 22-25.08)
- Expanded use cases, sequence diagrams (22.03.2003, inspection 22-25.08)
- Requirements validation (25.08.2003).

See also the later versions of General (sect. 1), User Requirements (sect. 4), and Project Plan (sect. 5).

2. Design (26.08–27.10.2003). The main artifacts: Design Specification:

- Architecture design (last draft 6.10.2003, final version 8.10).
- User interface design (last draft 6.10.2003, final version 10.10).
- Subsystems design (25.10.2003, analysis & inspection 25-27.10).
- Test plan & test cases design (29.10.2003, analysis & inspection 29-31.10).

3. Implementation (30.10–24.11.2003). also including unit testing.

4. Testing (26.11–10.12.2003). Integration and validation testing.

Conclusion of the Project (December, 2003). user manual completion, final customer validation.

5.8 Project Plan: Version 1.20

The waterfall model is chosen for the software process. The Customer may slightly change the requirements during “Requirements analysis” and “Design” phases; after that the requirements are frozen.

The start point of the Project is 7.07.2003; at this date the planning phase is initiated. The real development process starts 16.07.2003 when the first meeting of the project team is conducted. A working version of the required web system must be completed as the conclusion of the Project (November, 2003).

5.8.1 Phases

0. Initial planning (7.07–16.07.2003). Definition of the Project objectives and scope. Customer and Senior Manager discussing. Enrolling student software team. The skeleton of the maintenance document is produced (this one).
The main artifacts:

- Version 1.00 of the User Requirements (sect. 4.1),
- Version 1.00 of the Project Plan (sect. 5.1).

1. Requirements analysis (16.07–25.08.2003). Analysis of the problem domain and user requirements.
The main artifacts: Requirements Specification:

- Expanded User Requirements (13-15.08.2003, inspection 20.08)
- Glossary of the problem domain (15.08.2003, inspection 20.08).
- Conceptual model of the Problem domain (18.08.2003, inspection 20.08).
- High-level architecture (18.03.2003, inspections 20-22.08)
- High-level use cases, use case diagram (18.03.2003, inspection 20.08)
- System requirements (22.03.2003, inspections 22-25.08)
- Expanded use cases, sequence diagrams (22.03.2003, inspection 22-25.08)
- Requirements validation (25.08.2003).

See also the later versions of General (sect. 1), User Requirements (sect. 4), and Project Plan (sect. 5).

2. Design (26.08–27.10.2003). The main artifacts: Design Specification:

- Architecture design (last draft 6.10.2003, final version 8.10).
- User interface design (last draft 6.10.2003, final version 10.10).
- Subsystems design (25.10.2003, analysis & inspection 25-27.10).
- Test plan & test cases design (29.10.2003, analysis & inspection 29-31.10).

3. Unit implementation (30.10–28.11.2003). also including unit testing.

4. Integration implementation (28.11–7.12.2003). also includes integration testing.

5. Validation testing (8.12–15.12.2003).

Conclusion of the Project (December, 2003). user manual completion, final customer validation.

5.9 Project Plan: Version 1.21

The waterfall model is chosen for the software process. The Customer may slightly change the requirements during "Requirements analysis" and "Design" phases; after that the requirements are frozen.

The start point of the Project is 7.07.2003; at this date the planning phase is initiated. The real development process starts 16.07.2003 when the first meeting of the project team is conducted. A working version of the required web system must be completed as the conclusion of the Project (December, 2003).

5.9.1 Phases

0. Initial planning (7.07–16.07.2003). Definition of the Project objectives and scope. Customer and Senior Manager discussing. Enrolling student software team. The skeleton of the maintenance document is produced (this one).
The main artifacts:

- Version 1.00 of the User Requirements (sect. 4.1),
- Version 1.00 of the Project Plan (sect. 5.1).

1. Requirements analysis (16.07–25.08.2003). Analysis of the problem domain and user requirements.
The main artifacts: Requirements Specification:

- Expanded User Requirements (13-15.08.2003, inspection 20.08)
- Glossary of the problem domain (15.08.2003, inspection 20.08).
- Conceptual model of the Problem domain (18.08.2003, inspection 20.08).
- High-level architecture (18.03.2003, inspections 20-22.08)
- High-level use cases, use case diagram (18.03.2003, inspection 20.08)
- System requirements (22.03.2003, inspections 22-25.08)

- Expanded use cases, sequence diagrams (22.03.2003, inspection 22-25.08)
- Requirements validation (25.08.2003).

See also the later versions of General (sect. 1), User Requirements (sect. 4), and Project Plan (sect. 5).

2. Design (26.08–27.10.2003). The main artifacts: Design Specification:

- Architecture design (last draft 6.10.2003, final version 8.10).
- User interface design (last draft 6.10.2003, final version 10.10).
- Subsystems design (25.10.2003, analysis & inspection 25-27.10).
- Test plan & test cases design (29.10.2003, analysis & inspection 29-31.10).

3. Unit implementation (30.10–06.12.2003). also including unit testing.
The main artifacts:

- Implementation document
- Test execution document

4. Integration implementation (03.12–13.12.2003). also includes integration testing.
The main artifacts:

- **The Web-SynDic System**

5. Validation testing (13.12–20.12.2003). See Test execution document.

Conclusion of the Project (December, 2003). user manual completion, final customer validation.

5.10 Project Plan: Version 1.22

The waterfall model is chosen for the software process. The Customer may slightly change the requirements during "Requirements analysis" and "Design" phases; after that the requirements are frozen.

The start point of the Project is 7.07.2003; at this date the planning phase is initiated. The real development process starts 16.07.2003 when the first meeting of the project team is conducted. A working version of the required web system must be completed as the conclusion of the Project (December, 2003).

5.10.1 Phases

0. Initial planning (7.07–16.07.2003). Definition of the Project objectives and scope. Customer and Senior Manager discussing. Enrolling student software team. The skeleton of the maintenance document is produced (this one).
The main artifacts:

- Version 1.00 of the User Requirements (sect. 4.1),
- Version 1.00 of the Project Plan (sect. 5.1).

1. Requirements analysis (16.07–25.08.2003). Analysis of the problem domain and user requirements.
The main artifacts: Requirements Specification:

- Expanded User Requirements (13-15.08.2003, inspection 20.08)
- Glossary of the problem domain (15.08.2003, inspection 20.08).
- Conceptual model of the Problem domain (18.08.2003, inspection 20.08).
- High-level architecture (18.03.2003, inspections 20-22.08)
- High-level use cases, use case diagram (18.03.2003, inspection 20.08)
- System requirements (22.03.2003, inspections 22-25.08)
- Expanded use cases, sequence diagrams (22.03.2003, inspection 22-25.08)
- Requirements validation (25.08.2003).

See also the later versions of General (sect. 1), User Requirements (sect. 4), and Project Plan (sect. 5).

2. Design (26.08–27.10.2003). The main artifacts: Design Specification:

- Architecture design (last draft 6.10.2003, final version 8.10).
- User interface design (last draft 6.10.2003, final version 10.10).
- Subsystems design (25.10.2003, analysis & inspection 25-27.10).
- Test plan & test cases design (29.10.2003, analysis & inspection 29-31.10).

3. Unit implementation (30.10–06.12.2003). also including unit testing.
The main artifacts:

- Implementation document
- Test execution document

4. Integration implementation (03.12–20.12.2003). also includes integration testing.
The main artifacts:

- **The Web-SynDic System**

5. Validation testing (15.12–20.12.2003). See Test execution document.

Conclusion of the Project (December, 2003). user manual completion, final customer validation.

5.11 Project Plan: Version 2.00

Working version of the product is ready. It is not stable. System testing has been chosen to continue the process. It starts at 12.01.2004 and will be completed at May 2004.

Preparation to alpha-testing (12–15.01.2004).

1st stage of alpha testing with 3rd year students of Mathematical faculty (16-18.01.2004).

Analysis of discovered errors and debugging, the next version of Web-SynDic. Start to form the to-do list (16.01–10.02.2004).

2nd stage of alpha-testing with 2nd&3rd year students of Mathematical faculty (16–20.02.2004)

Analysis of discovered errors and debugging, the next version of Web-SynDic. Start to form the to-do list (16–22.02.2004).

3rd stage of alpha testing: Microsoft conference in SPb (24–25.02.2004).

Analysis of the results of alpha-testing. Work according with the to-do list, the next version of Web-SynDic (March–April 2004).

Publication of Web-SynDic, beta-testing (April–May 2004).

5.12 Project Plan: Version 2.10

Working version of the product is ready. It is not stable. System testing has been chosen to continue the process. It starts at 12.01.2004 and will be completed at May 2004.

Preparation to alpha-testing (12–15.01.2004).

1st stage of alpha testing with 3rd year students of Mathematical faculty (16-18.01.2004).

Analysis of discovered errors and debugging, the next version of Web-SynDic. Start to form the to-do list (16.01–10.02.2004).

3rd stage of alpha testing: Microsoft conference in SPb (16.02–5.03.2004).

Analysis of discovered errors and debugging, the next version of Web-SynDic. Start to form the to-do list (5–14.03.2004).

3nd stage of alpha-testing with 2nd&3rd year students of Mathematical faculty (10–20.03.2004)

Analysis of the results of alpha-testing. Work according with the to-do list, the next version of Web-SynDic (March–April 2004).

Publication of Web-SynDic, beta-testing (April–May 2004).

6 Development Process

6.1 Development History

7.07–16.07.2003 Yury Bogoyavlenskiy (Customer), Dmitry Korzun (Senior Manager). Initial planning of the Project. The base maintenance document is produced (sect.1, v.1.00; sect.2, v.1.00; sect.4, v.1.00; sect.5, v.1.00).

16.07.2003: Starting of the Project, the first meeting of the project team.

16.07.2003, evening: Web site and CVS server for the Project are installed at zeta.cs.karelia.ru. Thanks to Vadim Ponomarev. Access for team members and stockholders only. At this moment the CVS sever is under testing.

17.07.2003: Refinement of the ArgoUML tool by Mikhail Kryshen'. Now Russian letters are normally supported by ArgoUML (and some other minor but useful changes). The result can be found here.

18.07.2003: The first meeting for discussing the User requirements.

18.07.2003: Minor textual fixes in User Requirements (see sect. 4), 4.2).

21.07.2003: Meeting for starting the requirement analysis phase.

24.07.2003: Meeting for detailing the user requirements.

24.07.2003: Minor fixes in User Requirements (see sect. 4), 4.3).

24.07.2003: Customer birthday :–) The team tries to activate the process.

24.07.2003, evening: The CVS repository at zeta.cs.karelia.ru starts to be filled up. Thanks to Petr Semin. The team waits instructions from him.

25.07.2003: The templates for a functional requirement is released, see file `ur-template.tex` (expanded user requirement) and `sr-template.tex` (system requirement) in the doc template directory.

03.08.2003, 15:00 Team is leaving Petrozavodsk.

04.08.2003, 4:30 Team is arriving to Helsinki.

04.08.2003, day Different organizational issues.

12:30 Registration in the International Relations Office.

13:00 Receiving scholarships.

16:00 Talk of Timo Alanko (introduction).

16:30 Excursion in CS Department of Helsinki University.

17:00 Registration, computer network accounts, miscellaneous.

05.08.2003, 10:15–12:50 Intro lecture on SE (Juha Taina).

06.08.2003 The expert group (University of Helsinki) has an access to the Web page of the Project. We start to write the Requirements specification document. The template for a high-level use case is released, see file `uc_high-template.tex`

07.08.2003, day Introduction meetings with Finnish student teams (all team).

09:45 Digimon, interviewer is A. Salo.

11:30 Synkka, interviewer is P. Semin.

14:00 Kuvastin, interviewer is K. Kulakov.

08.08.2003 Customerinspects and changes General (see sect. 1, 1.2).

08.08.2003 Customerinspects and corrects User Requirements (see sect. 4, 4.4).

08.08.2003, day Introduction meetings with Finnish groups (all team) and lectures on SE.

09:00 Nippu, interviewer is A. Ananin.

10:15 Lecture on Project management (Juha Taina).

11:15 Juna, interviewer is M. Kryshen'.

12:15 Lecture on Project management (Juha Taina).

08.08.2003 The customer makes some corrections to his requirements (see section 4.4 for version 1.10 of User Requirements).

12.08.2003, day Meetings with Finnish groups.

10:00 Digimon, interviewer is A. Salo.

14:00 Juna, interviewer is M. Kryshen'.

14:00 Kuvastin, interviewer is K. Kulakov.

13.08.2003, 10:15–13:00 Lecture on Testing (Juha Taina).

14.08.2003, day Meetings with Finnish students groups.

12:00 Nippu, interviewer is A. Ananin.

14:00 Synkka, interviewer is P. Semin.

11.08.2003 Changes in Project Plan (see sect. 5, 5.2). Requirements phase is prolonged.

15.08.2003 Minor improvements of General (see sect. 1, 1.3). Senior Managerinspection.

17.08.2003 Detailing and modifying Project Plan (see sect. 5, 5.3). Senior Managerinspection.

17.08.2003 Elicitation and detailing of User Requirements (see sect. 4, 4.5). Senior Managerinspection.

18.08.2003 Russian branch of the Project web page is started: Maintenance document, Meetings chronology.

24.08.2003 Links to personal web pages were added in Project Team (see sect. 2, 2.2).

27.08.2003 Web-SynDic seminar: the team gives talks on its work and results. The slides can be found here.

29.08.2003, 11:00 The team leaves Helsinki.

05.09.2003, 14:00 Meeting with Customer. Discussion on future prospects.

08.09.2003 Design stage is started.

17.09.2003 From now on team works 4 hours a day in "dedicated" computer class (usually from 09:45 to 13:05).

20.09.2003 Updating and inspecting Project Plan (see sect. 5, 5.4). Senior Managerinspection.

25.09.2003 Current version of requirement specifications is considered to be perfect and frozen.

01.10.2003 Updating stages in Project Plan — correction according to delay (see sect. 5, 5.5). Senior Managerinspection.

20.10.2003 Updating stages in Project Plan — correction according to delay (see sect. 5, 5.6). Senior Managerinspection.

28.10.2003 Updating stages in Project Plan — correction according to delay (see sect. 5, 5.7). Senior Managerinspection.

29.10.2003 Implementation stage is started (unit implementation and testing).

17.11.2003 P. Semin and N. Rodchenkova are not in the team now; they, unfortunately, leave us; they cannot perform the required amount of work.

3.12.2003 Integration phase of implementation and testing is started. See the current version of the Web-SynDic system. Only AlgorithmServer subsystem (the most complex and important one) is at unit implementation phase.

6.12.2003 See copy of HTML 4.01 specification at the Web-SynDic project server.

6.2 Meetings

16.07.2003, 11:00–13:30. The first meeting of the project team.

Participants: Y. Bogoyavlenskiy (Customer), V. Ponomarev (Expert), D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager, Secretary), A. Salo (Senior Practitioner), A. Anan'in (Practitioner), M. Kryshen' (Practitioner).

Topics:

1. 11:10–12:06, Y. Bogoyavlenskiy, talk "Project objectives and Scope". Discussion.
2. 12:16–12:49, D. Korzun, talk "Problem domain of the Project" (see also section 3). Discussion.
3. 12:49–13:20, D. Korzun, talk "Organization of the process" (see also section 5). Discussion.
4. 13:20–13:30, D. Korzun, tasks assignment.

Task assignment:

1. Each team member. The Web-SynDic maintenance document: read and analyze, main attention to the user requirements. The first discussion on 18.07.2003 at 13:00.
2. P. Semin, K. Kulakov. CVS server: location (zeta), testing, access restrictions, repository structure. Discussion on 24.07.2003.
3. M. Kryshen', P. Semin. ArgoUML: general test working, required resources (Linux, Windows), coding problems (English, Russian). Discussion on 26.07.2003.

More details of meeting chronology.

18.07.2003, 13:00. Start discussing the user requirements.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager, Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner).

Topics: 1. 13:00–13:57, D. Korzun, lecture "Introduction to Software Engineering: the Product and the Process"

2. 14:04–14:50 Discussion on the User Requirements.

Task assignment: 1. Each team member. Analysis of each requirement: more precise definition, more detail specification, risk analysis, simplification, elicitation, priority, etc. (21.07.2003)

2. P. Semin. The CVS server: preliminary report on a current state (21.07.2003).

3. A. Salo. User registration: a review of available tools (24.07.2003).

More details of meeting chronology.

21.07.2003, 14:00. Start of requirements analysis.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager, Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner).

Topics: 1. 14:00–14:40, D. Korzun, lecture "Requirements Analysis"

2. 14:40–15:20 Detailed discussion on the User Requirements. Requirements classification and assignment among developers for further analysis.

Task assignment: 1. Each team member. Collecting problem domain terms for the glossary. Detailed analysis of the requirements and their specification using the given template. Preliminary theoretical study of "use cases".

2. K. Kulakov: draft of the glossary (24.07.2003).

3. P. Semin. The CVS server: report on a current state, a draft of server structure, and a brief manual for team members (24.07.2003).

4. M. Kryshen'. Requirements F5, F6 (24.07.2003).

5. A. Salo. Requirements F3, F4 (24.07.2003).

6. K. Kulakov, P. Semin. Requirements F1a–F2b (24.07.2003).

More details of meeting chronology.

24.07.2003, 13:30. Detailing the user requirements. Use cases.

Participants: V. Ponomarev (Expert), D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager, Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. 13:30–14:11, D. Korzun, lecture "Use cases analysis".

2. 13:11–14:20, General discussion on CVS, JDK, ArgoUML.

3. 14:20–15:11, Discussion on detailed specification of the user requirements.

Task assignment: 1. Each team member. Detailed specification of user requirements (using the template `ur-template.tex`).

2. A. Ananin. Preliminary list of use cases for Web-SynDic (26.07.2003).

3. K. Kulakov. Expansion of the problem domain glossary (elec.form).

4. P. Semin. A brief CVS manual for team members (26.07.2003).

More details of meeting chronology.

26.07.2003, 10:30. User and system requirements. Use cases. Organizational aspects.

Participants: Y. Bogoyavlenskiy (Customer), D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager, Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. 10:30–10:45, P. Syomin. General CVS manual for Web-SynDic.

2. 10:45–11:45, Discussion on detailed specification of the user requirements.

3. 11:45–12:00, K. Kulakov. Expanded version of glossary. A. Ananin. Draft of list of use cases.

4. 12:00–12:20, Y. Bogoyavlenskiy. General organizational aspects of project.

Task assignment: 1. Each team member. Specification of system requirements. (using the template `sr-template.tex`).

2. A. Ananin. Detailed list of use cases and the corresponding diagram for Web-SynDic.

3. A. Salo, M. Kryshen'. Review of tools for Web-SynDic to store and process statistical information about users.

4. D. Korzun, M. Kryshen'. Specification of view of report.

More details of meeting chronology.

01.08.2003, 12:00. Discussing about previously assigned tasks.

Participants: K. Kulakov (Technical Manager, Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. 12:02–12:17. System requirements: initial specifications.

2. 12:17–12:30. A. Ananin. Advanced list of use cases.

3. 12:30–12:44. A. Salo. Review of tools for project to process statistical data about users.
4. 12:44–12:55. M. Kryshen'. Specification of report view.

Task assignment: Each team member: continue accomplishing previously assigned tasks.

More details of meeting chronology.

05.08.2003, 13:30. Discussion on current tasks and goals of project. Organizational aspects.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: Discussion on current tasks and goals of project.

Task assignment:
1. Each team member: specification of system requirements.
2. Each team member: final version of user requirements.
3. Each team member: all specifications of requirements must be added to CVS repository before 14:00 06.08.2003.
4. A. Ananin: detailed list of use cases and corresponding diagrams for Web-SynDic.
5. Next lecture on SE (Juha Taina) will be on 08.08.2003, 10:15.

More details of meeting chronology.

06.08.2003, 14:00. Meetings with Finnish software teams. User requirement specifications. Use cases.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics:
1. Discussion on first meetings with the Finnish teams. Each team member has been assigned to one group.
2. Discussion and correction of user requirement specifications.
3. Discussion on use cases (A. Ananin).

Task assignment:
1. Each team member: expanded versions of user requirement specifications; system requirement specifications on CVS.
2. Each team member: work plan.
3. P. Semin: brief manual on CVS-server zeta.cs.karelia.ru.

4. A. Ananin: detailed list of use cases and corresponding diagrams for Web-SynDic.
5. First meetings with Finnish teams are assigned.

More details of meeting chronology.

07.08.2003, 11:00. User and system requirement specifications. System functions. Use cases.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics:
1. Discussion on glossary (K. Kulakov).
2. Discussion and correction of user requirement specifications and their design.
3. Discussion on system requirement specifications.
4. Precise list of all system function. Function assignment for each team member.
5. Discussion on use cases (A. Ananin). Use cases assignment for each team member.

Task assignment:
1. Each team member: final versions of user requirement specifications; system requirement specifications on CVS.
2. Each team member: work plan.
3. A. Ananin: list of use cases and corresponding diagrams for Web-SynDic.
4. K. Kulakov: expanded version of glossary.
5. P. Semin: diagram of approximate system architecture.

More details of meeting chronology.

11.08.2003, 12:25 Progress reports. User and system requirement specifications. Use cases.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics:
1. Discussion on glossary (K. Kulakov). Preliminary conceptual model.
2. Discussion and correction of system requirement specifications.
3. Use cases: main diagram, templates.

Task assignment:
1. Each team member: system requirement specifications; use cases specifications; terminology in specifications and glossary;

2. Each team member (except A. Ananin and M. Kryshen'): description of one similar system for solving NLDE system; A. Ananin and M. Kryshen': analysis of project "007".
3. K. Kulakov: glossary, preliminary thinking over conceptual model.

More details of meeting chronology.

13.08.2003, 13:45 Progress reports. High-level use cases. Reviews of ANLDE or LP/MILP programs. System requirement specifications. Problem domain.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics:
1. Discussion on conceptual model and glossary (K. Kulakov).
2. Expanded user requirements.
3. Discussion on high-level use cases.
4. Reviews of similar (ANLDE or LP/MILP) programs.
5. Review of documentation of "007" project (A. Ananin, M. Kryshen').

Task assignment:
1. Each team member: expanded user requirements; use cases specifications.
2. A. Ananin: expanded use cases diagram.
3. K. Kulakov: expanded conceptual model; changed glossary structure.
4. A. Salo: diagram of problem domain structure.
5. P. Semin: expanded diagram of system architecture.

More details of meeting chronology.

15.08.2003, 12:15 Discussion on previously assigned tasks and interviews.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics:
1. Discussion on the previous task assignment.
2. Results of interview Finnish teams (each team member).

Task assignment:
1. Each team member: prepare slides for the final meeting.
2. A. Ananin: 1) templates for personal html pages of team members; 2) all sequence diagrams for the use cases.

3. A. Salo: 1) brief description for LP/MIP algorithms (to be included in the appendix of the Requirements Specification); 2) comprehensive analysis of the problem domain and Web-SynDic initial architecture.
4. M. Kryshen': 1) definition of minimal hardware characteristics for a Web-SynDic server and for Web-SynDic client; 2) comprehensive analysis of requirements (expanded user and system) and high-level use cases.

More details of meeting chronology.

20.08.2003, 13:15 Discussion on previously task assignment.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics:
1. Discussion on the previous task assignment.
2. Discussion on use cases diagrams and UML-tools.

Task assignment:
1. Each team member: prepare slides for final meeting.
2. Each team member: description of use cases based on diagrams.
3. A. Ananin: complete sequence diagrams for all use cases.
4. P. Semin: remove empty directories in CVS repository.

More details of meeting chronology.

23.08.2003, 17:00 Progress report. Current tasks.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics:
1. Discussion on use cases descriptions and sequence diagrams. Discussion about finalizing current project stage.

Task assignment:
1. Each team member: prepare slides for final meeting.
2. A. Ananin and A. Salo: additional development of some sequence diagrams.
3. P. Semin: validation criteria.

More details of meeting chronology.

25.08.2003, 13:05 Over the requirement analysis stage.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Issues to determine: working hours, personal pages, architecture and use cases diagrams.

Task assignment: individual tasks for each team member.

More details of meeting chronology.

27.08.2003, 14:00 Preliminary discussion on design stage.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Discussion on design stage.

Task assignment: preliminary thinking over further tasks.

More details of meeting chronology.

08.09.2003, 14:00 Start of design stage.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Further tasks arrangement. Stages of design, descriptions.

Task assignment: 1. P. Semin, K. Kulakov: specification and correction of the system architecture.

2. M. Kryshen': user interface.
3. K. Kulakov, P. Semin: subsystems interface.
4. A. Salo, M. Kryshen', K. Kulakov: component level design.
5. A. Ananin, P. Semin: testing design.

More details of meeting chronology.

12.09.2003, 12:00 Advancing design process.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Discussion on detailed architecture issues.

2. Discussion on Web-SynDic interfaces.
3. General organizational issues.

Task assignment: 1. N. Rodchenkova: fixes in development history.

2. M. Kryshen', A. Ananin: detailed user interface design.
3. K. Kulakov, P. Semin: subsystems interface.
4. P. Semin, K. Kulakov: analysis and corrections of EUR, SR and UC.
5. P. Semin: sketch of Design Document.

More details of meeting chronology.

15.09.2003 Sketch file of Design Document and corresponding directory are accomplished on CVS.

15.09.2003, 12:00 Advancing design process.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Description of EUR, SR, UC analysis.

2. Detailed system architecture.
3. "Dedicated" computer class.
4. User interface.

Task assignment: 1. P. Semin, K. Kulakov: final detailed architecture and subsystems.

2. A. Salo, P. Semin, K. Kulakov: interfaces of subsystems.
3. A. Ananin, M. Kryshen': interface(s).

More details of meeting chronology.

17.09.2003, 09:55 Advancing design process.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Discussion on software for the computer class.

2. System architecture issues.
3. User interface issues.

Task assignment: 1. P. Semin, K. Kulakov, A. Salo: detailed system architecture.

2. K. Kulakov, P. Semin: EUR, SR, UC.
3. A. Ananin, M. Kryshen': user interface.

More details of meeting chronology.

20.09.2003, 11:30 Advancing design process.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), M. Kryshen' (Practitioner).

Topics: 1. Different organizational issues.

2. System architecture.
3. User interface.

Task assignment: 1. Each team member: working hours; personal web pages.

2. P. Semin, K. Kulakov, A. Salo: detailed system architecture: static structure model; draft of architecture design document; interface and dynamical models.
3. A. Ananin, M. Kryshen', K. Kulakov, A. Salo: user interface: client part and interface; input/output data; interface (draft) of corresponding section.
4. A. Ananin, M. Kryshen': draft of full user interface design document.

More details of meeting chronology.

22.09.2003, 12:30 Advancing design process.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), M. Kryshen' (Practitioner)

Topics: 1. EUR, UC and SR analysis.

2. Interface model. Behavioral model.
3. Structure of Design Document. Discussion on user interface.

Task assignment: 1. Each team member: working hours; personal web pages.

2. A. Ananin, A. Salo, M. Kryshen': input/output data control.
3. A. Salo: behavioral model.
4. K. Kulakov: fixes and risks in UC and EUR.
5. P. Semin: analysis and fixes in EUR, UC, SR.

More details of meeting chronology.

25.09.2003, 12:00 Advancing design process.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Requirement specifications. Current version is considered to be perfect and frozen.

2. System architecture.
3. User interface issues. Internal IO file formats.

Task assignment: 1. Each team member: personal web pages.

2. P. Semin: freeze current version of requirements specification.
3. N. Rodchenkova: complete translation of current version of requirements specification.
4. P. Semin, K. Kulakov, A. Salo: behavioral model.
5. A. Ananin, M. Kryshen': full input and output data file formats.

More details of meeting chronology.

27.09.2003, 12:30 Advancing design process.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Personal web pages.

2. Discussion on architecture.
3. User interface issues.

Task assignment: 1. Each team member: advance personal web pages.

2. P. Semin, K. Kulakov, A. Salo: complete behavioral model.
3. K. Kulakov: subsystems assignment.
4. A. Ananin, M. Kryshen': fill style conventions section, complete input and output formats.
5. A. Ananin, M. Kryshen': prepare notes about implementation.

More details of meeting chronology.

02.10.2003, 12:00 Advancing design process.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Architecture issues.

2. Subsystems assignment.
3. Test plan. User interface issues.

Task assignment: 1. A. Ananin, M. Kryshen': form generation, sequence processing.

2. A. Ananin, M. Kryshen': full User Interface Design Document, without "Introduction" section.
3. A. Salo, K. Kulakov: final Architecture Design Document.
4. P. Semin: test plan.
5. Each team member: SE-tools.

More details of meeting chronology.

06.10.2003, 10:30 Advancing design process.

Participants: D. Korzun (Senior Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Organizational issues.

2. Progress overview.
3. Subsystems assignment.

Task assignment: 1. Each team member: working hours.

2. Each team member: subsystems.
3. P. Semin: test plan.

More details of meeting chronology.

09.10.2003, 11:00 Advancing design process.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner).

Topics: 1. Subsystems Design.

2. Testing and Test Plan.
3. Different implementation issues.
4. Discussion on server.

Task assignment: 1. Each team member: performing current tasks.

2. A. Salo, M. Kryshen': discussion about server with Vadim Ponomarev.

More details of meeting chronology.

13.10.2003, 10:30 Advancing design process.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Subsystems Design.

2. Testing and Test Plan.

Task assignment: 1. Each team member: performing current tasks.

2. Each team member: working hours.
3. A. Salo, M. Kryshen': discussion about server with Vadim Ponomarev.

More details of meeting chronology.

16.10.2003, 11:30 Advancing design process.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Subsystems Design.

2. Testing and Test Plan.

Task assignment: 1. Each team member: performing current tasks.

2. A. Salo: activity statistics re-assigned.

More details of meeting chronology.

20.10.2003, 11:00 Advancing design process.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Organizational Issues.

2. Subsystems Design.

Task assignment: 1. Each team member: performing current tasks.

2. Each team member: working hours.

More details of meeting chronology.

25.10.2003, 11:30 Completing design process.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Subsystems Design.

Task assignment: 1. Each team member: performing current tasks.

2. A. Salo, A. Ananin, M. Kryshen': unit testing.
3. K. Kulakov, A. Salo, A. Ananin, M. Kryshen': algorithms description in every class specification; define integration testing combination.
4. Each team member: working hours.

More details of meeting chronology.

29.10.2003, 12:00 Starting implementation stage.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Organizational issues.

2. Implementation problems assignment:
   (a) M. Kryshen': web server;
   (b) A. Salo: translators and activity statistics;
   (c) P. Semin: Test Plan and Test Execution Document;
   (d) A. Ananin: management and data store;
   (e) K. Kulakov: analysis and concordance of all sections of Design Document.

Task assignment: 1. M. Kryshen: web server issues.

2. P. Semin: complete Test Plan.
3. A. Salo: grammar for translators.
4. K. Kulakov: analysis of Design Document.
5. M. Kryshen', A. Ananin, A. Salo: start implementation of assigned problems.

More details of meeting chronology.

05.11.2003, 12:00 Implementation.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Organizational issues.

2. Progress report:
   (a) M. Kryshen': web server;
   (b) A. Salo: translators;
   (c) A. Ananin: management and data store;
   (d) P. Semin: Test Plan and Test Execution Document;
   (e) K. Kulakov: analysis and concordance of all sections of Design Document.

Task assignment: 1. M. Kryshen: web server issues.

2. A. Salo: complete translators.
3. A. Ananin: complete subsystem classes.
4. M. Kryshen, A. Salo, A. Ananin: unit testing.
5. K. Kulakov: complete analysis of Design Document.
6. N. Rodchenkova: translation of "Design Document".

More details of meeting chronology.

10.11.2003, 11:00 Implementation.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Progress report:
   (a) M. Kryshen': web server;
   (b) A. Salo: translators;
   (c) A. Ananin: management and data store;
   (d) K. Kulakov: analysis Design Document.

Task assignment: 1. M. Kryshen: web server issues.

2. A. Salo: complete translators.
3. A. Ananin: complete subsystem classes.
4. M. Kryshen, A. Salo, A. Ananin: unit testing.
5. K. Kulakov: finalize Design Document.

More details of meeting chronology.

13.11.2003, 12:00 Implementation.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), P. Semin (Technical Manager), N. Rodchenkova (Secretary), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

Topics: 1. Progress report:

(a) M. Kryshen': web server;

(b) A. Salo: translators;

(c) A. Ananin: management and data store;

(d) K. Kulakov: algorithm server;

(e) P. Semin: User Guide.

Task assignment: 1. M. Kryshen, A. Ananin, A. Salo, K. Kulakov: Implementation Document; unit testing.

2. Each team member: continue to perform current tasks.

3. Each team member: working hours and main results.

More details of meeting chronology.

17.11.2003, 11:00 Implementation

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

11:00 P.Semin and N.Rodchenkova are excluded from the team.

11:05 Progress report:

1. A. Ananin: unit testing & data store;
2. M. Kryshen': unit testing & web server;
3. K. Kulakov: generators & solvers, execution files;

11:20 Priority: 1) unit testing just after implementing a unit, 2) comments during coding, 3) satisfaction to comment standards and conventions.

11:25 Task assignment:

1. Each team member: activate the implementation, but the priority must be satisfied.
2. Each team member: fill working hours and main results.

11:30 Conclusion.

More details of meeting chronology.

22.11.2003, 12:00 Implementation.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

12:00 A.Salo reports on implementation of ANLDE translator. For one ANLDE system, the translator works properly (tested). Assignment: Class ANLDERequest must be implemented as soon as possible (required by another subsystems). Translator for ANLDE systems set must be implemented and tested; select an algorithm for less memory consumption.

12:10 M.Kryshen' reports on implementation of the web server (JSP). Only 3 pages have been completed to this moment, 11 have not. Assigned: force the implementation, but unit testing must not be skipped.

12:15 A.Ananin reports on implementation of subsystem "Data store". Only management has not been completed to this moment. Assignment: continue management implementation (3 methods) and its unit testing.

12:20 K.Kulakov reports on implementation of subsystem "Algorithm server". Two generators are implemented and tested. Assignment: implementation and testing of 3 solvers (anlde, slopes, lp_solve).

12:25 Discussing of the non-implemented units: parsers for generator and solver outcomes, spoolers and statistics. Supposed, they will be assigned to Kulakov, Salo and Ananin. Kryshen is totally concentrates on the web server.

12:30 Conclusion.

27.11.2003, 12:00 Implementation.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

12:00 M.Kryshen reports on the web server implementation. The most part of web forms is ready (ANLDE processing forms are only not implemented). No servlets to this moment. Web pages for errors anu user notification are not implemented. Assignment: finish web forms and pages, produce a list of subtasks to implement (with priorities). Be ready to give some of these tasks to other implementors.

12:05 A.Salo reports on the ANLDE parsers. The second ANLDE parser (ANLDE systems set) is almost ready, only the last tests must be executed. Assignment: Execute the last tests. Move to implementation of the Statistics subsystem.

12:10 A.Anan'in reports on his implementation. All assigned subsystems are ready. Assignment: Start the implementation of translators for generator's and solver's output.

12:11 A problem of ineffective usage of memory for outcome translator is discussed. Let a less effective realization be than better but more difficult for implementation one.

12:15 K.Kulakov reports on algorithm server implementation. Two generators work correctly. The solvers will work very soon. Assignment: Finishing solvers implementation. Start to implement Spoolers.

12:20 Discussion on session processing. Who (or what) is responsible for object Statistics? This will be done at integration stage by tomcat features.

12:25 Preliminary discussion on integration. We'll use the bottom-up approach. Assignment: Each developer makes a list of combinations, started with his subsystem and ended with the web server. M.Kryshen makes a list of large combinations passed through the web server.

12:30 Conclusion.

01.12.2003, 10:00 Implementation.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

M.Kryshen reports on the web server implementation. All forms are implemented and tested. Readiness for integration is not completed (management only).
Assignment: Prepare the web server for the integration (Management, ANLDEParser, ActivityStatistics, AlgorithmServer). Wednesday is a milestone.

A.Salo reports on the ActivityStatistics implementation. The implementation is completed, but no unit testing, no test plan, no implementation notes.
Assignment: Fill in all related documents (test plan, implementation, test execution). Move to integration with DataStore. After that implementation of ANLDE2TXT parser. Thursday is a milestone.

K.Kulakov reports on the implementation of the AlgorithmServer. To this moment class CheckSolution has been implemented and started to test.
Assignment: finish testing (test plan, test execution and implementation document). Move to implementation of Spooler classes. Saturday is a milestone.

A.Ananin reports on the implementation of Generator and Solver outcomes parsers. GeneratorOutcomeParser has been implemented and started to test.
Assignment: Finish unit testing of GeneratorOutcomeParser and move to implementation of SolverOutcomeParser (2 parsers). Saturday is a milestone.

Conclusion . We should be in hurry.

06.12.2003, 12:00 Implementation (integration).

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), A. Salo (Senior Practitioner), M. Kryshen' (Practitioner), A. Ananin (Practitioner).

M.Kryshen reports on the web server integration. Integration with ANLDEParser and Management goes on.
Assignment: Continue the integration. For management testing, a test user must be registered. Monday is a milestone.

A.Salo reports on the ANLDEParser integration with the web server. There are some problems with HTML. The process of their solutions goes on.
Assignment: Fill test plan and test execution. Monday is a milestone.

K.Kulakov reports on the implementation of the AlgorithmServer. Generator and GeneratorOutputParser are integrated and tested.
Assignment: Move to integration of Solver and SolverOutcomeParser. Monday is a milestone.

A.Ananin reports on the implementation Solver outcome parsers. The parsers were implemented and tested.
Assignment: Move to integration of Management—DataStore (test plan and test execution). Then move to AlgorithmServer: integration of cluster GeneratorSpooler—Generator. Wednesday is a milestone.

A.Ananin starts discussing the problem of automatic generation of test instances of ANLDE class. Result: Ananin may use Generator and GeneratorOutputParser for generating test instances.

A.Salo starts discussing on email of the system administrator (for noting). Result: XML must be used by the web server to store the data.

Conclusion with discussing the next project—distributed project with Helsinki University. The general scope and the latest news.

13.12.2003, 11:45 Implementation (integration).

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), A. Salo (Senior Practitioner), A. Ananin (Practitioner).

K.Kulakov reports on cluster SolverSpooler—*. Test execution has been started but not yet finished. class CheckSolution will be integrated soon.
Assignment: Complete the integration among with the test execution. Wednesday is a milestone.

A.Anan'in reports on cluster GeneratorSpooler—*. Test plan is ready. The test execution starts.
Assignment: Check the test plan (inspection). Complete the test execution. Monday is a milestone.

A.Salo reports on the ANLDEParser and ActivityStatistics (integration). The uncovered problem with ANLDEParser (coefficient limit checking) has been fixed. Test plan for WebServer—ActivityStatistics—* is ready.
Assignment: Start test execution for cluster WebServer—ActivityStatistics—*. Ask M.Kryshen about UI issues (activity statistics table). Coordinate complex tests with M.Kryshen. Wednesday is a milestone.

Conclusion with discussing the next project—distributed project with Helsinki University. The latest news about accommodation and travel. The customer wishes to see us today at 14:30.

20.12.2003, 15:00 Finishing the Project. Developers move to pass exams (PetrSU). K.Kulakov and D.Korzun will trim the Web-SynDic system during December and January. Now the system in alpha-testing.

24.01.2004, 16:20 End of first alpha-testing.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), A. Salo (Senior Practitioner), A. Ananin (Practitioner), M. Kryshen(Practitioner).

K.Kulakov reports on new bugs of Web-SynDic system.
Assignment: Report on alpha-testing. Start to analyse java code and check it for corresponding conventions to javadocs. Add 'Main result' section on the personal web-page. Note all tested ANLDE systems with differences in solving process using various algorithms.

A.Anan'in Assignment: Start user guide section with a 'Registration' and 'Session' parts. Start to analyse java code and check it for corresponding conventions to javadocs. Add 'Main result' section on the personal web-page. Note all tested ANLDE systems with differences in solving process. Count lines of java code and summary volume of jsp pages.

A.Salo Assignment: Start to fix the bugs in the Web-SynDic. Start user guide section with a 'Administration' part. Start to analyse java code and check it for corresponding conventions to javadocs. Add 'Main result' section on the personal web-page. Note all tested ANLDE systems with differences in solving process using various algorithms.

M.Kryshen Assignment: Start to fix the bugs in the Web-SynDic. Start to analyse java code and check it for corresponding conventions to javadocs. Add 'Main result' section on the personal web-page. Note all tested ANLDE systems with differences in solving process using various algorithms.

Conclusion with discussing the next project—distributed project with Helsinki University. The latest news about accommodation and travel.

26.01.2004, 15:00 End of first alpha-testing.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), A. Salo (Senior Practitioner), A. Ananin (Practitioner), M. Kryshen(Practitioner).

K.Kulakov discusses the external algorithm's monitoring problem and the lp_solver perfomance problem.
Assignment: Report on alpha-testing. Start to analyse java code and check it for corresponding conventions to javadocs. Add 'Main result' section on the personal web-page. Note all tested ANLDE systems with differences in solving process using various algorithms. Fix the perfomance problem with the lp_solver. Fix the monitoring of the external alogorithm's processes.

A.Anan'in Assignment: Start user guide section with a 'Registration' and 'Session' parts. Start to analyse java code and check it for corresponding conventions to javadocs. Add 'Main result' section on the personal web-page. Note all tested ANLDE systems with differences in solving process. Count lines of java code and summary volume of jsp pages.

A.Salo Assignment: Start to fix the bugs in the Web-SynDic. Start user guide section with a 'Administration' part. Start to analyse java code and check it for corresponding conventions to javadocs. Add 'Main result' section on the personal web-page. Note all tested ANLDE systems with differences in solving process using various algorithms.

M.Kryshen Assigment: Start to fix the bugs in the Web-SynDic. Start to analyse java code and check it for corresponding conventions to javadocs. Add 'Main result' section on the personal web-page. Note all tested ANLDE systems with differences in solving process using various algorithms.

Conclusion with the decision to make Saturday to be a rest-day and Sunday to be a working-day.

Meeting finished at 15:21.

28.01.2004, 16:20 End of first alpha-testing.

Participants: D. Korzun (Senior Manager), K. Kulakov (Technical Manager), A. Salo (Senior Practitioner), A. Ananin (Practitioner), M. Kryshen(Practitioner).

M.Kryshen reports on fixed bug of the Web-system
Assigment: Start to fix next bugs in the Web-SynDic. Continue analyse java code and check it for corresponding conventions to javadocs. Add 'Main result' section on the personal web-page. Note all tested ANLDE systems with differences in solving process using various algorithms.

A.Anan'in reports on counting some metrics of the Web-SynDic (lines of java code, volume of jsp-pages)
Assignment: Start user guide section with a 'Registration' and 'Session' parts. Continue analyse java code and check it for corresponding conventions to javadocs. Add 'Main result' section on the personal web-page. Note all tested ANLDE systems with differences in solving process.

A.Salo Assignment: Start to fix the bugs in the Web-SynDic. Start user guide section with a 'Administration' part. Continue to analyse java code and check it for corresponding conventions to javadocs. Add 'Main result' section on the personal web-page. Note all tested ANLDE systems with differences in solving process using various algorithms.

K.Kulakov reports on list of Web-SynDic features
Assignment: Report on alpha-testing. Continue analyse java code and check it for corresponding conventions to javadocs. Add 'Main result' section on the personal web-page. Note all tested ANLDE systems with differences in solving process using various algorithms.

Conclusion with discussing the next project—distributed project with Helsinki University. The latest news about accommodation and travel.

6.3 Working Hours

Kirill Kulakov

	Date	Time	Activity
W01	16.07	11:00–13:23	Meeting
	13.07	15:00–16:45	Learning requirements
	18.07	13:00–14:50	Meeting
	19.07	13:00–18:25	Work with requirements
	20.07	13:00–19:15	Work with glossary
Sum	4	9:38	
W02	21.07	14:00–15:20	Meeting
	22.07	14:10–16:05	Work with user requirements
	24.07	13:30–15:11	Meeting
	25.07	11:00–12:40	Work with user requirements
	25.07	15:15–13:00	Work with glossary
	26.07	10:30–12:25	Meeting
	26:07	12:10–13:20	Work with system requirements
Sum	5	11:06	
W03	01.08	12:00–13:00	Meeting
	02.08	13:00–14:50	Work with system requirements
Sum	2	02:50	
W04	05.08	10:15–12:50	Intro lecture on SE (Juha Taina)
	05.08	13:30–13:50	Meeting
	05.08	14:00–15:30	Work with user requirements
	05.08	15:35–18:00	Work with system requirements
	06.08	14:00–15:00	Meeting
	06.08	15:05–16:00	Work with user requirements
	06.08	16:05–18:00	Work with system requirements
	07.08	11:00–12:25	Meeting
	07.08	12:30–13:30	Work with user requirements
	07.08	13:35–14:05	Work with system requirements
	08.08	10:15–11:15	Lecture on Project Management (Juha Taina)
	08.08	12:15–13:15	Lecture on Project Management (Juha Taina)
	10.08	11:00–13:45	Work with system requirements
	10.08	20:00–21:00	Filling working hours
Sum	5	19:20	
W05	11.08	10:00–10:30	Work with system requirements and working hours
	11.08	12:25–13:25	Meeting
	11.08	13:25–14:30	Work with glossary
	11.08	14:30–18:00	Work with conceptual model
	12.08	12:00–13:15	Work with conceptual model

	12.08	13:50–20:00	Work with conceptual model
	12.08	20:30–23:30	Work with system requirements
	13.08	10:00–13:00	Juha Taina’s lecture
	13.08	13:43–15:25	Meeting
	13.08	15:25–16:00	Work with glossary
	13.08	16:00–18:00	Work with user and system requirements
	14.08	11:00–18:00	Work with conceptual model
	15.08	12:15–13:05	Meeting
Sum	5	29:37	
W06	18.08	14:30–15:00	Filling working hours
	18.08	15:00–15:30	Make personal page
	19.08	15:30–16:00	Meeting (D. Korzun, myself and P. Semin)
	19.08	16:00–18:00	Preparing documentation of SE course
	20.08	13:15–13:45	Meeting
	20.08	13:45–17:00	Preparing documentation of SE course
	20.08	17:00–20:00	Learning sequence diagrams
	23.08	17:00–17:30	Meeting
	24.08	19:00–19:30	Make translation of personal page
	24.08	19:30–20:00	Filling working hours
	24.08	20:00–21:40	Work with use cases
Sum	5	13:25	
W07	25.08	13:00–13:20	Meeting
	25.08	13:20–20:00	Work with system requirements
	25.08	20:00–21:00	Work with use cases
	25.08	21:00–22:30	Work with slides
	27.08	09:00–11:00	Seminar
	27.08	12:10–12:35	Meeting
	27.08	13:00–19:00	Making documentation for SE course
	28.08	11:00–20:30	Making documentation for SE course
Sum	3	27:25	
W08	00.09	00:00–00:00	
Sum	0	00:00	
W09	08.09	12:25–13:00	Meeting
	10.09	15:15–18:00	Work with SE course

	11.09	14:00–18:40	Work with SE course
	12.09	12:00–12:45	Meeting
	13.09	10:00–15:00	Learning Requirements Specification
	13.09	16:00–19:00	Work with Architecture
Sum	5	16:45	
W10	15.09	12:00–12:50	Meeting
	17.09	09:55–10:40	Meeting
	17.09	10:40–13:05	work with requirements
	17.09	15:00–16:00	Making documentation for SE course
	17.09	18:00–19:00	Making documentation for SE course
	18.09	09:45–13:05	work with requirements
	18.09	13:30–15:15	work with requirements
	18.09	15:15–20:15	work with SE course
	19.09	09:45–11:20	work with requirements
	19.09	11:20–13:05	work with server architecture
	20.09	09:45–10:20	Making documentation for SE course
	20.09	10:20–11:35	work with subsystem interface
	20.09	11:35–12:00	Meeting
	20.09	12:00–13:05	work with subsystem interface
Sum	5	22:35	
W11	22.09	09:45–12:30	work with subsystem interface
	22.09	12:30–13:05	meeting
	22.09	16:00–18:40	Making documentation for SE course
	24.09	09:45–13:05	work with personal page and architecture
	25.09	09:45–13:05	Work with requirements specification
	25.09	15:00–18:40	work with SE course
	26.09	09:45–13:05	Work with behavioral model
	27.09	11:30–12:30	Work with behavioral model
	27.09	12:30–13:00	Meeting
	27.09	13:00–15:15	Work with behavioral model
Sum	5	23:25	
W12	29.09	09:45–13:05	work with behavioral model
	01.10	09:45–13:05	work with behavioral model
	01.10	15:15–17:00	prepare documentation for SE course

	02.10	09:45–12:00	work with behavioral model
	02.10	12:00–12:30	meeting
	02.10	12:30–13:05	work with system architecture
	02.10	15:15–20:15	work with SE course
	03.10	09:45–13:05	work with system architecture
	04.10	09:45–13:05	work with system architecture
Sum	5	23:25	
W13	08.10	09:45–13:05	work with server subsystems interface
	08.10	15:15–17:00	prepare documentation for SE course
	09.10	09:45–11:00	work with server subsystems interface
	09.10	11:00–11:35	meeting
	09.10	11:35–13:05	work with algorithm server architecture
	09.10	15:15–18:40	work with SE course
	10.10	09:45–13:05	work with algorithm server architecture
	11.10	11:20–15:15	work with algorithm server architecture
Sum	4	19:05	
W14	13.10	09:45–10:30	work with server subsystems interface
	13.10	10:30–11:05	meeting
	13.10	11:05–13:05	work with server subsystems interface
	15.10	09:45–13:05	work with server subsystems interface
	15.10	15:15–17:00	prepare documentation for SE course
	16.10	09:45–11:30	work with algorithm server architecture
	16.10	11:30–12:05	meeting
	16.10	12:05–13:05	work with algorithm server architecture
	16.10	15:15–20:15	work with SE course
	17.10	09:45–13:05	work with algorithm server architecture
	18.10	09:45–13:05	work with tests for algorithm server
Sum	5	23:25	
W15	20.10	09:45–11:00	work with tests for algorithm server
	20.10	11:00–11:30	meeting
	20.10	11:30–13:05	work with tests for algorithm server
	22.10	09:45–13:05	work with tests for algorithm server
	23.10	09:45–13:05	work with algorithm server architecture
	23.10	15:15–18:40	work with SE course

	24.10	09:45–13:05	work with tests for algorithm server
	25.10	09:45–13:05	work with algorithm server architecture
Sum	5	20:05	
W16	27.10	09:45–13:05	work with algorithm server architecture
	29.10	09:45–13:05	work with algorithm server architecture
	29.10	12:00–12:30	meeting
	29.10	12:30–13:05	design specification analysis
	29.10	15:15–17:00	prepare documentation for SE course
	30.10	09:45–13:05	design specification analysis
	30.10	15:15–20:15	work with SE course
	31.10	09:45–13:05	design specification analysis
	01.11	09:45–13:05	design specification analysis
Sum	5	23:25	
W17	03.11	09:45–13:05	design specification analysis
	05.11	09:45–12:00	design specification analysis
	05.11	12:00–12:30	meeting
	05.11	12:30–13:05	design specification analysis
	05.11	15:15–17:00	prepare documentation for SE course
	06.11	09:45–13:05	design specification analysis
	06.11	15:15–18:40	work with SE course
	09.11	13:30–16:30	design specification analysis
	09.11	16:30–17:30	meeting with customer
Sum	4	19:10	
W18	10.11	09:45–11:05	design specification analysis
	10.11	11:05–11:30	meeting
	10.11	11:30–13:05	design specification analysis
	12.11	09:45–13:05	prepare external algorithms
	12.11	15:15–17:00	prepare documentation for SE course
	13.11	09:45–12:00	prepare external algorithms
	13.11	12:00–12:25	meeting
	13.11	12:25–13:05	prepare external algorithms
	13.11	15:15–20:15	work with SE course
	14.11	09:45–13:05	prepare external algorithms
	15.11	09:45–13:05	implementation generator classes

Sum	5	23:25	
W19	17.11	09:45–11:00	implementation generator classes
	17.11	11:00–11:30	meeting
	17.11	11:30–13:05	implementation generator classes
	19.11	09:45–13:05	implementation generator classes
	19.11	15:15–17:00	prepare documentation for SE course
	20.11	09.45–13:05	implementation and start testing generator classes
	20.11	15:15–18:40	work with SE course
	21.11	09:45–13:05	implementation and testing generator classes
	22.11	09:45–12:00	implementation and testing generator classes
	22.11	12:00–12:30	meeting
	22.11	12:30–13:05	implementation solver classes
Sum	5	21:50	
W20	24.11	09:45–13:05	implementing classes for solver class
	26.11	09:45–13:05	implementing and testing solving classes
	26.11	15:15–17:00	prepare documentation for SE course
	27.11	09:45–12:00	testing solving classes
	27.11	12:00–12:30	meeting
	27.11	12:30–13:05	testing solving classes
	27.11	15:15–20:15	work with SE course
	28.11	09:45–13:05	implementing and testing solving classes
	29.11	09:45–13:05	implementing and testing solving classes
Sum	5	23:25	
W21	01.12	09:45–10:00	implementing classes for solver class
	01.12	10:00–10:40	meeting
	01.12	10:40–13:05	implementing and testing algorithm server classes
	03.12	10:00–13:05	implementing GeneratorSpooler class
	04.12	09:45–13:05	implementing SolverSpooler class
	05.12	09:45–13:05	implementing SolverSpooler class
	06.12	09:45–13:05	integration Generator–GeneratorOutputParser
	06.12	16:00–18:40	implementing methods toString() for all classes in algorithm server
Sum	5	19:05	
W22	08.12	09:45–13:05	integration Solver–SolverOutputParser

	10.12	09:45–13:05	integration and testing Solver—SolverOutputParser
	11.12	11:05–13:05	testing SolverSpooler—Solver—*
	13.12	11:30–14:30	testing SolverSpooler—Solver—*
Sum	4	11:40	
W23	15.12	09:45–12:05	testing SolverSpooler—*—CheckSolution
	15.12	12:05–13:05	Integration web server and SolverSpooler
	17.12	09:45–13:05	Integration ad testing web server and SolverSpooler
	17.12	11:00–15:15	Integration ad testing web server and SolverSpooler
	18.12	09:45–13:05	Integration ad testing web server and SolverSpooler
	19.12	09:45–13:05	Integration ad testing web server and SolverSpooler
	20.12	09:45–13:05	Integration ad testing web server and SolverSpooler
Sum	5	20:55	
W24	27.12	14:00–16:20	Fixed same bags in algorithm server and web forms
Sum	1	02:20	
W25	29.12	15:00–16:00	Fixed solving a set of ANLDE systems
Sum	1	01:00	
W26	05.01	10:00–10:30	Fixed Check solutions class
	05.01	12:00–13:30	Fixed ANLDE system characteristics class
	08.01	10:00–11:00	Fixed solvers and generators classes
	08.01	11:00–12:00	Fixed Solver metrics class
Sum	2	04:00	
W27	15.01	10:00–12:00	Added additional server characteristics and fixed session manager class
	17.01	10:00–11:00	Fixed notes dispatcher class
	17.01	12:00–13:00	Fixed algorithms classes
Sum	2	04:00	
W28	19.01	11:00–12:30	Fixed blank algorithm files
Sum	1	01:30	
W29	28.01	15:00–19:00	Adding alpha-testing
Sum	1	04:00	
W30	02.02	15:00–16:00	Modifying lp-solver class
	02.02	16:00–17:30	Fixing process class
	02.02	17:30–18:00	Fixing alpha-testing
Sum	1	03:00	

Month	Time spent (dd; hh:mm)	Process phase
July	09; 20:44	Requirement analysis
August	20; 92:37	Requirement analysis
September	16; 66:05	Design
October	22; 102:45	Design
November	20; 91:10	Design, Implementation
December	16; 55:00	Implementation, Testing
January	07; 16:30	Implementation, Testing
Total	110; 446:51	

Natalia Rodchenkova

	Date	Time	Activity
W04	04.08	12:30–17:30	Different organizational issues. CS Department of Helsinki university
	05.08	10:15–12:50	Intro lecture on SE (Juha Taina)
	05.08	13:30–13:50	Meeting
	05.08	14:00–15:00	Meeting logs
	06.08	14:00–15:00	Meeting
	06.08	15:00–16:00	Meeting logs
	06.08	16:00–17:00	Translating file process.tex
	07.08	11:00–12:25	Meeting
	07.08	12:30–13:30	Meeting logs
	08.08	10:15–11:15	Lecture on Project Management (Juha Taina)
	08.08	12:15–13:15	Lecture on Project Management (Juha Taina)
	08.08	14:00–15:00	Translating file process.tex
Sum	5	17:20	
W05	11.08	12:25–13:25	Meeting
	11.08	13:30–14:30	Meeting logs
	13.08	10:15–13:00	Lecture on Testing (Juha Taina)
	13.08	13:45–15:25	Meeting
	13.08	15:30–16:30	Meeting logs
	14.08	13:45–14:15	Filling working hours
	14.08	15:00–17:00	Translating documentation in russian
	15.08	12:15–13:05	Meeting
	15.08	13:20–14:00	Meeting logs
	15.08	14:05–15:05	Translating documentation in russian
	15.08	15:05–15:10	Filling working hours
Sum	4	11:50	
W06	19.08	15:00–16:00	Making personal page in english
	19.08	16:00–16:30	Making personal page in russian
	19.08	16:10–16:30	Filling working hours
	20.08	13:15–13:45	Meeting
	20.08	14:00–15:00	Meeting logs in english and russian
	20.08	15:10–15:30	Filling working hours
	20.08	16:00–17:00	Work with file process.tex
	21.08	14:00–14:30	Advancing personal pages
	21.08	14:30–14:50	Filling working hours
	23.08	17:00–17:25	Meeting
	23.08	22:00–23:00	Meeting logs in english and russian
	23.08	23:10–23:20	Filling working hours
Sum	4	07:05	
W07	25.08	13:00–13:20	Meeting
	25.08	13:20–14:00	Meeting logs in english and russian
	25.08	14:00–20:00	Translations. Different fixes. Urgent tasks of leader.
	26.08	16:00–21:00	Preparing slides and speech for final seminar.
	27.08	09:00–11:00	Seminar
	27.08	12:10–12:35	Meeting
	27.08	12:35–13:30	Meeting logs in english and russian
	27.08	17:00–20:00	Finalizing current documentation.
	27.08	20:00–21:00	Filling working hours.
Sum	3	19:20	
W09	08.09	12:25–13:00	Meeting
	08.09	13:00–14:00	Meeting logs in english and russian
	09.09	15:00–19:00	Minor fixes in documentation
	12.09	12:00–12:45	Meeting
	25.08	12:40–13:30	Meeting logs in english and russian
	12.09	13:00–16:00	Translation of documentation of Requirement Analysis stage

	13.09	13:00–16:00	Translation of documentation of Requirement Analysis stage
Sum	4	13:00	
W10	15.09	12:00–12:50	Meeting
	17.09	09:55–10:40	Meeting
	17.09	10:40–13:05	Meeting logs in english and russian. Revision and translation of documentation of Requirement Analysis stage. Filling working hours.
	18.09	09:45–13:05	Inspection of changed Requirement Analysis documentation. Partial translations.
	19.09	09:45–13:05	Partial translations of documentation of Requirement Analysis stage.
	20.09	11:20–12:00	Meeting.
	20.09	12:00–13:20	Meeting logs in english and russian.
Sum	5	12:40	
W11	22.09	09:45–12:30	Partial translations of documentation of Requirement Analysis stage.
	22.09	12:30–13:00	Meeting.
	22.09	13:00–14:00	Meeting logs in english and russian.
	24.09	09:45–13:00	Translations of fixes in documentation of Requirement Analysis stage. Filling working hours.
	25.09	09:45–12:00	Translations of fixes in documentation of Requirement Analysis stage. Fixing personal web-page.
	25.09	12:00–13:00	Meeting.
	25.09	13:00–14:00	Meeting logs in english and russian.
	26.09	09:45–13:00	Translations of documentation of Requirement Analysis stage.
	27.09	09:45–12:30	Completing of translation of requirements.
	27.09	12:30–13:00	Meeting.
	27.09	13:00–14:00	Meeting logs in english and russian.
Sum	6	19:15	
W12	29.09	09:45–13:00	...
	01.10	09:45–13:00	...
	02.10	09:45–13:00	...
	03.10	09:45–13:00	...

	04.10	09:45–13:00	...
	05.10	09:45–13:00	...
Sum	5	00:00	

Month	**Time spent (dd; hh:mm)**	**Process phase**
July	00; 00:00	Requirement analysis
August	16; 55:35	Requirement analysis
September	00; 00:00	Design
October	00; 00:00	
November	00; 00:00	
Total	00; 00:00	

Petr Semin

	Date	**Time**	**Activity**
W01	14.07	11:00–16:00	Preparing review of UML tools
	16.07	11:00–13:25	Meeting
	16.07	13:25–14:30	Meeting log
	16.07	14:30–16:20	Requirements analysis
	18.07	13:00–14:50	Meeting
	18.07	13:50–15:50	Meeting log
	18.07	16:00–17:00	Requirements analysis
Sum	3	12:10	
W02	21.07	14:00–15:20	Meeting
	21.07	15:20–16:20	Meeting log
	21.07	16:30–17:00	CVS-server structure
	24.07	11:00–13:30	Learning CVS system
	24.07	13:30–15:10	Meeting
	24.07	18:00–19:30	Managing CVS server
	24.07	19:30–20:30	Meeting log
	25.07	11:00–14:00	Brief CVS manual
	26.07	10:30–12:30	Meeting
	26.07	12:30–13:30	Meeting log
	26:07	13:30–15:00	Learning UML tool ArgoUML
Sum	4	17:00	

W03	01.08	12:00–13:00	Meeting
	01.08	13:00–14:00	Managing CVS repository
Sum	1	02:00	
W04	04.08	14:00–17:00	CS Department of Helsinki university
	05.08	10:15–12:50	Intro lecture on SE (Juha Taina)
	05.08	13:30–14:00	Meeting
	06.08	14:00–15:00	Meeting
	06.08	15:00–16:00	Preparing system requirements
	07.08	11:00–12:30	Meeting
	07.08	12:30–14:00	Preparing diagram of Web-SynDic general architecture
	07.08	14:00–16:00	Managing CVS repository. Meetings log corrections.
	07.08	16:00–18:00	Different patterns added
	08.08	10:15–11:15	Lecture on Project Management (Juha Taina)
	08.08	12:15–13:15	Lecture on Project Management (Juha Taina)
	10.08	19:00–20:00	Preparing system requirements
	10.08	21:00–23:00	Filling working hours
Sum	5	19:05	
W05	11.08	12:20–13:30	Meeting
	11.08	13:30–14:30	Fixing system requirements, use cases
	11.08	14:30–15:30	Poseidon and ArgoUML tools, correcting architecture diagram.
	11.08	18:00–20:00	Different minor fixes, advancing and checking CVS manual
	12.08	12:00–12:30	CVS manual finalized and sent to customer.
	12.08	12:30–14:00	Minor fixes in russian branch. Working with patterns of working hours.
	12.08	14:00–15:00	ArgoUML and architecture diagram.
	13.08	10:15–13:00	Lecture on Testing (Juha Taina)
	13.08	13:45–15:25	Meeting
	13.08	15:30–17:30	Use cases. System requirements. Finalizing architecture diagram.
	13.08	20:30–21:00	Correcting patterns of working hours. Minor fixes on CVS.
	14.08	12:00–13:30	Changing working hours according to changed leader's demands.
	14.08	14:00–16:00	Meeting with Synkka.

	15.08	12:15–13:05	Meeting
	15.08	13:30–14:00	Different minor fixes in CVS
	15.08	18:00–20:00	Different translations
Sum	5	22:25	
W06	18.08	20:00–23:30	Translations of last changes in documentation.
	19.08	11:00–12:00	Preparing SE tasks and docs
	19.08	15:30–16:00	Meeting (D. Korzun, myself and K. Kulakov)
	19.08	16:30–17:00	Filling working hours
	20.08	13:15–13:45	Meeting
	20.08	14:00–16:00	Writing and translating personal Web-page
	20.08	16:00–18:00	Learning sequence diagrams. Writing textual descriptions for use cases.
	21.08	12:00–14:00	Preparing documentation for SE course.
	21.08	18:00–19:30	Learning Java-tools.
	21.08	20:00–23:00	Preparing documentation for SE course.
	23.08	17:00–17:30	Meeting
	23.08	21:00–22:20	Minor fixes in documentation and translations
	23.08	22:20–23:30	Filling working hours
	23.08	20:30–23:50	Prepating validation criteria
Sum	5	22:20	
W07	25.08	11:00–17:00	Urgent corrections and fixes of leader.
	25.08	17:00–20:30	Preparing slides for final meeting.
	25.08	20:30–23:00	Final version of validation criteria.
	26.08	17:00–19:00	Finalizing slides and speech for final meeting.
	27.08	09:00–11:00	Seminar
	27.08	12:10–12:35	Final Helsinki meeting.
	27.08	13:00–14:00	Filling working hours.
	28.08	15:00–20:30	Checking tasks for PetrSU SE course.
Sum	4	22:55	
W09	08.09	12:25–13:00	Meeting.
	08.09	14:00–17:00	System architecture. Component level design.
	10.09	10:30–12:00	Checking EUR, SR, UC.
	11.09	16:00–20:00	PetrSU SE course.
	12.09	12:00–12:45	Meeting.

	12.09	12:45–14:00	Discussion about SE course.
	12.09	17:00–19:00	PetrSU SE course.
	14.09	14:00–15:00	Writing sketch of Design Document.
	14.09	15:00–17:00	Minor fixes in documentation.
Sum	5	15:40	
W10	15.09	12:00–12:50	Meeting.
	16.09	20:00–22:00	Advancing sketch of Design Document.
	17.09	09:45–13:05	Installing software. Filling working hours. Advancing system architecture. Different organizational issues on SE course.
	18.09	10:15–13:05	Advancing system architecture. Different textual comments and fixes. Filling working hours.
	19.09	10:30–13:05	Installing and learning UML tools. Collaboration diagrams and use cases. Preparing to unify system architecture sub-diagrams. Minor fixes in Design Document structure.
	19.09	13:30–18:30	PetrSU SE course.
	20.09	11:20–12:00	Meeting.
	20.09	12:00–14:00	Unifying system architecture sub-diagrams. Preparing for collaboration diagrams.
Sum	6	19:15	
W11	22.09	09:45–12:30	Static structure architecture model. Fixing Design Document.
	22.09	12:30–13:00	Meeting.
	24.09	09:45–13:00	Analysis of EUR, UC, SR. Filling working hours.
	25.09	09:45–12:00	Analysis of system architecture. Fixing personal web-page.
	25.09	12:00–13:00	Meeting.
	25.09	17:20–20:20	PetrSU SE course.
	26.09	09:45–13:00	Behavioral model.
	26.09	15:15–18:45	PetrSU SE course.
	27.09	10:30–12:30	Behavioral model. Collaboration diagrams.
	27.09	12:30–13:00	Meeting.
Sum	6	22:00	
W12	29.09	09:45–13:00	Behavioral model. Collaboration diagrams. Textual descriptions.
	01.10	09:45–13:00	Behavioral model. Textual descriptions.
	02.10	09:45–13:00	. . .

	03.10	09:45–13:00	. . .
	04.10	10:30–13:00	. . .
	04.10	15:15–18:45	PetrSU SE course.
Sum	5	00:00	

Month	Time spent (dd; hh:mm)	Process phase
July	07; 29:00	Requirement analysis
August	20; 88:45	Requirement analysis
September	00; 00:00	Design
October	00; 00:00	
November	00; 00:00	
Total	00; 00:00	

Andrey Salo

	Date	Time	Activity
W01	16.07	11:00–13:25	Meeting
	16.07	20:00–21:00	Reading UML documentation
	17.07	20:00–21:00	Reading Java documentation
	18.07	13:00–14:50	Meeting
	18.07	20:00–21:00	Reading Java documentation
	19.07	20:00–21:00	Reading Java documentation
Sum	6	08:15	
W02	21.07	14:00–15:20	Meeting
	21.07	21:00–23:00	Working with ArgoUML
	22.07	20:00–23:00	Reading CVS documentation
	23.07	20:30–22:00	Requirements analysis
	24.07	13:30–15:10	Meeting
	25.07	20:00–22:30	Reading Java documentation
	26.07	10:30–12:30	Meeting
	26.07	20:00–21:00	Reading UML documentation
	26.07	21:00–22:00	Working with ArgoUML
Sum	5	15:00	
W03	29.07	20:00–21:00	Reading Java documentation
	30.07	20:00–21:30	Reading Java documentation

	31.07	20:00–21:00	Requirements analysis
	01.08	12:00–13:00	Meeting
Sum	4	04:30	
W04	05.08	10:15–12:50	Intro lecture on SE (Juha Taina)
	05.08	13:30–14:00	Meeting
	05.08	14:00–17:00	CS Department of Helsinki university
	06.08	14:00–15:00	Meeting
	06.08	15:00–18:00	Preparing user requirements
	07.08	11:00–12:30	Meeting
	07.08	12:30–13:00	Preparing user requirements
	07.08	16:00–19:00	Working with Tomcat
	07.08	19:00–20:00	Reading JAAS documentation
	08.08	10:15–11:15	Lecture on Project Management (Juha Taina)
	08.08	12:15–13:15	Lecture on Project Management (Juha Taina)
	08.08	13:30–15:00	Reading JAAS documentation
	08.08	15:00–16:00	Working with Tomcat
	10.08	11:00–14:00	Preparing system requirements
Sum	5	23:35	
W05	11.08	11:00–12:15	Reading JAAS documentation
	11.08	12:20–13:30	Meeting
	11:08	22:00–23:00	Preparing to the interview
	12.08	10:00–11:00	Interview
	12.08	15:00–17:00	Working with user requirements
	12.08	17:00–18:30	Working with use cases
	13.08	10:15–13:00	Lecture on Testing (Juha Taina)
	13.08	13:45–15:25	Meeting
	14.08	10:00–11:30	Working on problem domain diagram
	14.08	11.30–12:00	Working on user requirements
	14.08	13:00–16:00	Working with Tomcat
	15.08	12:15–13:05	Meeting
	15.08	16:00–19:00	Working with Tomcat
	16.08	17:00–19:00	Reading bonsaiG documentation
	17.08	13:00–17:00	Working with external algorithms
Sum	7	27:10	

W06	18.08	14:00–16:00	Working with external algorithms
	18.08	16:00–17:00	Meeting
	18.08	20:00–22:30	Working with external algorithms
	19.08	16:00–18:30	Working with external algorithms
	19.08	18:30–19:00	Working with personal page
	20.08	11:00–13:00	Problem domain and architecture analysis
	20.08	13:15–13:45	Meeting
	20.08	20:00–23:00	Working with external algorithms
	21.08	21:30–23:00	Working with bonsaiG
	23.08	12:30–13:30	Working with personal page
	23.08	14:00–17:00	Reading Java documentation
	23.08	17:00–17:25	Meeting
	24.08	20:00–23:00	Filling working hours
	24.08	23:00–23:30	Working with personal page
Sum	6	23:25	
W07	25.08	00:00–02:10	Working with Tomcat
	25.08	13:00–13:15	Meeting
	25.08	13:30–19:00	Working with ArgoUML
	25.08	20:00–21:30	Reading Java servlets documentation
	26.08	13:00–17:30	Writing slides for presentation
	27.08	09:00–11:00	Presentation
	28.08	19:00–21:00	Reading documentation
Sum	4	17:55	
W09	08.09	12:00–13:00	Meeting
	09.09	21:00–23:00	Working with Java servlets
	11.09	11:30–13:00	Architecture design
	11.09	20:30–22:00	Client part design
	12.09	12:00–13:00	Meeting
	13.09	14:00–16:00	Reading documentation
Sum	5	09:00	
W10	15.09	12:00–12:50	Meeting
	16.09	20:30–23:00	Reading documentation
	17.09	09:50–10:40	Meeting
	17.09	10:40–13:05	Client part design

	17.09	22:00–00:00	Working with ArgoUML
	18.09	09:45–13:05	Preparing client part diagram
	19.09	10:30–14:30	Preparing client part description
	20.09	13:05–13:30	Meeting
	20.09	13:30–15:05	Filling working hours
	21.09	18:00–20:30	Reading documentation
Sum	7	20:25	
W11	22.09	09:45–11:30	Working with ArgoUML
	23.09	19:30–22:00	Reading UML documentation
	24.09	09:45–10:45	Discussing data representation
	24.09	10:45–13:05	Client part analysis
	24.09	22:30–00:00	Reading UML documentation
	25.09	09:45–12:00	Preparing client part description
	25.09	12:00–12:30	Meeting
	25.09	12:30–13:05	Preparing collaboration diagrams
	25.09	19:00–22:00	Reading UML documentation
	26.09	09:45–13:05	Preparing collaboration diagrams
	26.09	22:30–00:00	Reading UML documentation
	27.09	12:00–12:30	Architecture analysis
	27.09	12:30–13:00	Meeting
	27.09	13:00–15:05	Preparing collaboration diagrams
Sum	6	23:20	
W12	29.09	09:45–11:30	Architecture analysis
	30.09	20:00–22:00	Reading UML documentation
	01.10	09:45–13:05	Working on behavioral model
	02.10	09:45–12:00	Working on behavioral model
	02.10	12:00–12:30	Meeting
	02.10	12:30–13:05	Working on architecture
	03.10	09:45–13:05	Architecture design document
	04.10	11:30–15:05	Architecture design document
Sum	6	17:20	
W13	06.10	09:45–10:30	Working on architecture
	06.10	10:30–10:45	Meeting
	06.10	10:45–11:30	Subsystems design

	07.10	19:00–22:30	Session subsystem design
	08.10	09:45–13:05	Session subsystem design
	09.10	09:45–11:00	Session subsystem design
	09.10	11:00–11:35	Meeting
	09.10	11:35–13:05	Subsystems design
	10.10	09:45–13:05	Subsystems design
	11.10	11:30–16:00	Working on testing plan
Sum	6	19:45	
W14	13.10	09:45–10:30	Session subsystem design
	13.10	10:30–11:05	Meeting
	13.10	11:05–11:30	Subsystems design
	14.10	19:45–22:30	Reading Java documentation
	15.10	09:45–13:05	Session subsystem design
	16.10	09:45–11:30	Session manager design
	16.10	11:30–12:05	Meeting
	16.10	12:05–13:05	Session manager design
	17.10	09:45–13:05	Session manager design
	18.10	11:30–15:30	Activity statistics subsystem design
Sum	6	18:30	
W15	20.10	09:45–11:00	Activity statistics subsystem design
	20.10	11:00–11:30	Meeting
	21.10	19:30–23:00	Reading UML documentation
	22.10	09:45–13:05	Activity statistics subsystem design
	23.10	09:45–13:05	Activity statistics subsystem design
	24.10	09:45–13:05	Activity statistics subsystem design
	25.10	11:30–16:00	Working on activity statistics and session manager subsystems
Sum	6	19:45	
W16	27.10	09:45–11:30	Activity statistics subsystem design
	28.10	18:45–22:30	Reading Java documentation
	29.10	09:45–12:00	Writing unit testing for session manager
	29.10	12:00–12:30	Meeting
	29.10	12:30–13:05	Writing unit testing for session manager
	30.10	09:45–13:05	Implementing translators
	31.10	09:45–13:05	Implementing translators

	01.11	11:30–17:00	Implementing translators
Sum	6	21:00	
W17	03.11	09:45–11:30	Implementing translators
	04.11	18:00–23:00	Reading Java documentation
	05.11	09:45–12:00	Implementing translators
	05.11	12:00–12:30	Meeting
	05.11	12:30–13:05	Implementing translators
	06.11	09:45–13:05	Implementing translators
	07.11	09:45–13:05	Implementing translators
	08.11	11:30–17:00	Implementing translators
Sum	6	22:15	
W18	10.11	09:45–11:00	Implementing translators
	10.11	11:00–11:30	Meeting
	11.11	18:00–22:30	Reading Java documentation
	12.11	09:45–13:05	Implementing translators
	13.11	09:45–12:00	Implementing translators
	13.11	12:00–12:25	Meeting
	13.11	12:25–13:05	Implementing translators
	14.11	09:45–13:05	Implementing translators
	15.11	11:30–16:00	Testing translators
Sum	6	20:45	
W19	17.11	09:45–11:00	Testing translators
	17.11	11:00–11:30	Meeting
	18.11	18:00–23:00	Reading Java documentation
	19.11	09:45–13:05	Testing translators
	20.11	09:45–13:05	Writing implementation document
	21.11	09:40–13:05	Writing implementation document
	22.11	11:30–12:00	Testing translators
	22.11	12:00–12:30	Meeting
	22.11	12:30–16:00	Testing translators
Sum	6	21:15	
W20	24.11	09:45–12:15	ANLDE system set parser implementation
	25.11	20:00–22:00	ANLDE system set parser implementation
	26.11	09:45–13:05	ANLDE system set parser testing
	27.11	09:45–12:00	ANLDE system set parser testing
	27.11	12:00–12:30	Meeting
	27.11	12:30–13:05	ANLDE system set parser testing
	27.11	19:00–20:30	ANLDE system set parser testing
	28.11	09:45–15:30	Activity statistics implementation
	29.11	11:30–14:50	Activity statistics implementation
	29.11	14:50–15:40	Filling working hours
Sum	6	22:35	
W21	01.12	09:45–10:00	Filling working hours
	01.12	10:00–10:40	Meeting
	01.12	10:40–11:40	Activity statistics testing plan
	03.12	09:45–13:05	Activity statistics test cases & test execution
	03.12	15:30–18:30	Text representation of ANLDE objects
	04.12	09:45–13:05	Web server and ANLDE parser integration
	05.12	09:45–13:05	Web server and ANLDE parser integration & testing
	05.12	14:30–16:30	Web server and ANLDE parser (for system set) integration
	06.12	11:30–12:00	Web server and ANLDE parser testing
	06.12	12:00–12:40	Meeting
	06.12	12:40–15:05	Web server and ANLDE parser testing
Sum	5	20:30	
W22	08.12	09:45–11:30	Activity statistics and web server integration
	09.12	22:00–23:30	Fixing ANLDE parsers (error messages)
	10.12	09:45–13:05	Fixing ANLDE parsers (error messages)
	11.12	09:45–13:05	Web server and activity statistics integration test plan
	13.12	11:30–14:30	Web server and activity statistics integration
Sum	5	12:55	
W23	15.12	09:45–11:30	Web server and activity statistics integration
	15.12	14:00–16:30	Web server and activity statistics integration
	17.12	11:30–13:05	Web server and activity statistics test execution
	18.12	09:45–11:30	Web server and activity statistics test execution
	18.12	13:30–15:00	Fixing problems with null ip addresses
	19.12	09:45–13:05	Fixing problems with sessions; fixing problems with byaccj; filling test execution for statistics

	20.12	11:30–17:00	Web server and activity statistics further integration
Sum	5	17:55	
W26	05.01	12:00–15:00	Changes in session manager and activity statistics
Sum	1	03:00	
W27	13.01	15:00–19:00	Changes in notes dispatcher and activity statistics
	14.01	12:00–15:00	Changes in notes dispatcher and activity statistics
	15.01	14:00–17:00	Changes in management and notes dispatcher
	16.01	16:00–17:00	Changes in parser
Sum	4	11:00	
W28	24.01	16:20–17:00	Meeting
Sum	1	00:40	
W29	26.01	15:00–15:30	Meeting
	26.01	15:30–18:00	Working on user guide
	28.01	16:20–17:00	Meeting
	30.01	15:00–18:00	Working on user guide
Sum	3	06:40	
W30	02.01	16:00–16:30	Meeting
	02.01	16:30–18:00	Filling working hours
	04.01	16:00–20:00	Filling working hours
	06.02	15:00–16:00	Working on personal page
	06.02	16:00–18:30	Filling working hours
Sum	3+	08:30+	

Month	Time spent (dd; hh:mm)	Process phase
July	14; 26:45	Requirement analysis
August	23; 93:05	Requirement analysis
September	20; 56:30	Design
October	27; 87:05	Design
November	25; 92:20	Implementation, testing
December	15; 51:20	Implementation, testing
January	09; 21:10	Alpha-testing
February	03+; 08:30+	Alpha-testing
Total	146+; 436:45+	

Andrey Ananin

	Date	Time	Activity
W01	16.07	11:00–13:25	Meeting
	16.07	20:30–22:00	Reading maintenance document
	17.07	15:00–17:30	Testing ArgoUML
	18.07	15:15–16:15	Reading user's requirements
	19.07		was absent for valid reasons
	20.07		was absent for valid reasons
Sum	3	07:25	
W02	21.07		was absent for valid reasons
	22.07		was absent for valid reasons
	23.07		was absent for valid reasons
	24.07	13.30–15.10	Meeting
	25.07	20.00–22.40	Learning use cases
	25.07	22.40–23.40	Creating Web-SynDic use case diagram
	26.07	10.30–12.20	Meeting
	27.07	18.15–22.00	Learning UML
Sum	3	10:45	
W03	28.07	14.55–18.00	Learning CVS
	30.07	18.10–20.00	Developing use case diagram.
	01.08	12.00–13.00	Meeting
	01.08	15.15–17.00	Testing UML tools
	02.08	19.25–21.00	Developing use case diagram
Sum	4	09:15	
W04	05.08	13:30–14:00	Meeting
	05.08	14:00–17:00	CS Department of Helsinki university
	06.08	14:00–15:00	Meeting
	06.08	12.30–16.15	Writing use case diagram
	07.08	12.00–15.00	Developing use case diagram
	07.08	19.10–21.30	Developing system requirements
	07.08	11:00–12:30	Meeting OR
	08.08	10:15–11:15	Lecture on Project Management (Juha Taina)
	08.08	12:15–13:15	Lecture on Project Management (Juha Taina)
	08.08	14.00–15.00	Developing use case diagram
Sum	4	18:05	

W05	10.08	13.00–16.30	Fill system requirements pattern
	10.08	16.30–16.40	Adding system requirements on CVS
	10.08	16.35–17.00	Fill high use case pattern
	11.08	12:20–13:30	Meeting
	13.08	10:15–13:00	Lecture on Testing (Juha Taina)
	13.08	13:45–15:25	Meeting
	13.08	12:00–13:00	Interview with group ”Nippu”
	15.08	12:15–13:05	Meeting
	15.08	16:00–19:30	Making pattern for personal pages
	16.08	13:30–14:30	Reading UML
	16.08	17:25–19:30	Reading UML (sequence diagrams)
	16.08	19:30–22:30	Making sequence diagrams (first five)
Sum	5	21:05	
W06	17.08	13:35–16:00	Developing sequence diagrams
	17.08	20:00–21:30	Developing sequence diagrams
	18.08	16:00–17:00	Meeting
	18.08	17:00–18:30	Searching tools for converting ps to eps
	19.08	18:25–19:30	Learning UML
	19.08	21:15–23:30	Making sequence diagrams (last four)
	20.08	13:15–13:45	Meeting
	20.08	15:00–16:00	Developing sequence diagrams
	21.08	14:15–15:50	Developing sequence diagrams
	21.08	19:30–21:00	Making description of use case
	22.08	16:30–19:00	Analysing and changing some sequence diagrams
	23.08	16:00–17:00	Making and adding to repository Russian version of personal page
	23.08	17:00–17:25	Meeting
Sum	7	19:15	
W07	24.08	15:00–17:00	Making slides for presentation
	24.08	19:00–21:00	Checking and developing sequence diagrams
	25.08	13:00–13:15	Meeting
	25.08	14:00–15:35	Making presentation slides
	25.08	16:00–18:15	Add textual description to the use cases
	25.08	18:15–21:00	Making a new expanded user requirement

	27.08	09:00–11:00	Presentation
Sum	3	12:50	
W09	08.09	12:00–13:00	Meeting
	09.09	20:15–23:45	User interface design
	10.09	22:00–23:00	First notes about testing in design phase
	12.09	12:00–13:00	Meeting
Sum	4	06:30	
W10	15.09	12:00–12:40	Meeting
	16.09	21:25–00:05	Making textual description for the forms
	17.09	09:45–10:45	Meeting
	17.09	10:45–13:05	Editing textual description, user interface design
	18.09	09:45–11:00	Detailed forms
	18.09	11:00–13:05	Detailed sequence diagrams with corresponding forms
	19.09	09:45–13:05	Changed sequence diagrams, user interface design
Sum	5	14:20	
W11	24.09	09:45–12:05	Detailed sequence diagrams with corresponding forms
	24.09	12:05–13:05	Discussing data formats and form ”Notes”
	24.09	20:25–23:00	User interface design
	25.09	09:45–12:00	Discussing form ”Notes”, user interface design
	25.09	12:00–12:30	Meeting
	25.09	12:30–13:05	Change description about ”Notes” form
	26.09	09:45–13:05	Describe input data format
	27.09	09:45–12:30	Describe output data format
	27.09	12:30–13:00	Meeting
Sum	5	15:55	
W12	29.09	09:45–12:00	Output data format
	29.09	12:00–13:05	Changing forms
	01.10	09:45–13:05	Changing input and output data format, start preparing notes about implementation
	02.10	09:45–12:00	Changing style conventions section
	02.10	12:00–12:30	Meeting
	02.10	12:30–13:05	Changing ”Notes” form
	03.10	09:45–12:05	Changing form generation
	03.10	12:05–13:05	Sequence processing

	04.10	09:45–13:05	Changing User Interface Design Document without introduction
Sum	5	16:40	
W13	06.10	09:45–10:30	Changing User Interface Design Document
	06.10	10:30–10:45	Meeting
	06.10	10:45–13:05	Changing Design document
	08.10	09:45–13:05	Changing sequence processing
	09.10		Was absent for a valid reason
	10.10	09:45–13:05	Starting data store subsystem
	11.10	09:45–13:05	Continue creating data store subsystem
Sum	4	13:20	
W14	13.10	09:45–10:30	Changing data store subsystem
	13.10	10:30–11:05	Meeting
	13.10	11:05–13:05	Changing data store subsystem
	15.10	09:45–13:05	Changing data store subsystem
	16.10	09:45–11:30	Changing sequence diagrams
	16.10	11:30–12:05	Meeting
	16.10	12:05–13:05	Changing data store subsystem
	17.10	09:45–13:05	Creating data store subsystem diagram
	18.10	09:45–13:05	Changing data store subsystem description
Sum	5	16:40	
W15	20.10	09:45–11:00	Changing data store subsystem diagram
	20.10	11:00–11:30	Meeting
	20.10	11:30–13:05	Changing data store subsystem description
	22.10	09:45–13:05	Changing data store subsystem diagram and description
	23.10	09:45–13:05	Changing formats
	24.10	09:45–13:05	Changing and adding data store subsystem description
	25.10	09:45–11:30	Starting management subsystem
	25.10	11:30–12:00	Meeting
	25.10	11:30–13:05	Changing management subsystem
Sum	5	16:40	
W16	27.10	09:45–12:00	Adding algorithms description
	27.10	12:00–13:05	Discussing integration testing combinations
	29.10	09:45–12:00	Changing data store subsystem diagram and description

	29.10	12:00–13:05	Changing data store subsystem diagram and description
	30.10	09:45–12:05	Changing management subsystem diagram
	30.10	12:05–13:05	Changing management subsystem description
	31.10	09:45–13:05	Starting implementation of assigned subsystems
	01.11	09:45–13:05	Realizing data store subsystem
Sum	5	16:40	
W17	03.11	09:45–13:05	Realizing data store subsystem
	05.11	09:45–12:00	Realizing data store classes (DefaultsLimitsStore, UserProfile, Statistics)
	05.11	12:00–12:30	Meeting
	05.11	12:30–13:05	Creating unit tests
	06.11		Was absent for a valid reason
	07.11		Holiday
	08.11		Holiday
	09.11	09:45–13:05	Starting to realize management subsystem classes
Sum	3	10:00	
W18	10.11	09:45–11:00	Realizing management subsystem
	10.11	11:00–11:30	Meeting
	10.11	11:30–13:05	Realizing management subsystem and some statistic classes
	12.11	09:45–13:05	Changing management and data store subsystem classes
	13.11	09:45–12:00	Creating unit tests
	13.11	12:00–12:25	Meeting
	13.11	12:25–13:05	Changing management classes
	14.11	09:45–12:00	Writing Implementation document
	14.11	12:00–13:05	Changing javadoc
	15.11	09:45–13:05	Starting test execution phase
Sum	5	16:40	
W19	17.11	09:45–11:00	Creating execution tests for data store subsystem
	17.11	11:00–11:30	Meeting
	17.11	11:30–13:05	Changing execution tests
	19.11	09:45–13:05	Changing execution tests for data store subsystem
	20.11	09:45–13:05	Adding execution tests
	21.11	09:45–13:05	Adding execution tests for management subsystem
	22.11	09:45–12:00	Adding execution tests for management subsystem

	22.11	12:00–12:30	Meeting
	22.11	12:30–13:05	Changing management subsystem classes and filling test execution document
Sum	5	16:40	
W20	24.11	09:45–11:30	Changing javadoc
	24.11	11:30–13:05	Making Management class
	26.11	09:45–11:30	Realizing Management(function sendNotes())
	26.11	11:30–13:05	Learning Java Mail API
	27.11	09:45–12:00	Fixing minors
	27.11	12:00–12:30	Meeting
	27.11	12:30–13:05	Implementing OutputGeneratorSpooler class
	28.11	09:45–13:05	Implementing OutputGeneratorSpooler class and adding execution tests
	29.11	09:45–13:05	Implementing OutputGeneratorSpooler class and adding execution tests
Sum	5	16:40	
W21	01.12	09:45–10:00	Cvs updating
	01.12	10:00–10:40	Meeting
	01.12	10:40–13:05	Adding tests, changing tests, ending realization of GeneratorOutputPaser class
	03.12	09:45–13:05	Realization of BaseSolverOutputParser class, adding execution tests
	04.12	09:45–11:30	Was absent in valid reason
	04.12	11:30–13:05	Realizing Lp-solveOutputParser class, adding execution tests
	06.12	09:45–12:00	Realizing Lp-solveOutputParser class, adding execution tests
	06.12	12:00–12:30	Meeting
	06.12	12:30–13:05	Realizing Management class, function sendNotes(), adding execution tests
Sum	4	13:20	
W22	08.12	09:45–13:05	Making tests for Spooler-Generator-* branch
	10.12	12:00–13:05	Making test plan for Spooler-Generator-* branch
	11.12	09:45–11:30	Making test plan for Spooler-Generator-* branch
	11.12	12:00–13:05	Adding some tests for StatisticsStore class, adding execution tests for Spooler-Generator-* branch

	13.12	11:30–14:30	Making tests (test plan, execution tests) for for Spooler-Generator-* branch
Sum	4	10:15	
W23	15.12	11:20–13:05	Making validation tests for Session
	17.12	09:45–13:05	Making validation tests for Session, Log In and Sending a note test cases
	18.12	10:00–13:05	Making validation tests for Manage users, Manage default limits, Registration, Get statistics test cases
	19.12	09:45–13:05	Making execution tests for processing ANLDE systems (generating)
	20.12	11:30–13:05	Making execution tests for processing ANLDE systems (solving) Happy new year and good luck in exams!!!...8)
Sum	5	13:05	
W24	24.01	16:20–18:00	Meeting
	26.01	15:00–15:30	Meeting
	27.01	15:00–16:00	Counting and checking code metrics
	28.01	16:20–17:00	Meeting
	29.01	17:00–18:00	Making Registration section in User Guide
Sum	5	04:50	
W25	03.02	16:00–16:30	Changing Registration section in User Guide
	04.02	15:30–17:00	Checking java code
	04.02	17:00–18:00	Checking and calculating working hours
	05.02	15:00–17:00	Adding Results section in the personal page
Sum	3	05:00	
W26	25.02	15:15–18:30	Creating reference manual sections
	26.02	15:15–17:30	Creating reference manual section
	26.02	17:30–18:30	Calculating metrics
	27.02	13:30–15:00	Calculating metrics
	27.02	15:15–17:00	Calculating metrics
Sum	3	09:45	

Month	**Time spent (dd; hh:mm)**	**Process phase**
July	12; 23:05	Requirement analysis
August	21; 74:35	Requirement analysis
September	15; 40:05	Design
October	22; 73:20	Design
November	19; 63:20	Implementation and testing
December	13; 36:40	Implementation and testing
January	05; 04:50	Alpha-testing
February	05+; 14:45+	...
Total	113+; 330:40+	

Mikhail Kryshen

	Date	**Time**	**Activity**
W01	16.07	11:00–13:25	Meeting
	16.07	22:55–23:15	Reading documentation
	17.07	18:30–22:10	Testing and patching ArgoUML
	18.07	13:00–14:50	Meeting
Sum	3	08:15	
W02	21.07	14:00–15:20	Meeting
	22.07	20:30–21:00	Requirements analysis
	24.07	13:30–15:10	Meeting
	25.07	20:00–20:35	Learning documentation tools (TeX)
	26.07	10:30–12:25	Meeting
	27.07	20:00–20:40	Learning session management in servlets
Sum	6	06:40	
W03	01.08	00:05–00:40	Requirements analysis
	01.08	12:00–13:00	Meeting
Sum	2	01:35	
W04	05.08	10:15–12:50	Lecture
	05.08	13:30–13:50	Meeting
	05.08	14:00–18:00	Requirements analysis
	06.08	11:00–13:50	Requirements analysis
	06.08	14:00–15:00	Meeting
	07.08	11:00–12:25	Meeting
	08.08	10:15–11:15	Lecture
	08.08	11:20–12:10	Requirements analysis
	08.08	12:15–13:15	Lecture
	10.08	11:00–16:00	Requirements analysis
Sum	5	20:00	
W05	11.08	10:30–12:20	Requirements analysis
	11.08	12:25–13:25	Meeting
	12.08	13:30–13:55	Requirements analysis
	12.08	14:00–15:00	Interview
	13.08	10:15–13:00	Lecture
	13.08	13:05–13:40	Requirements analysis
	13.08	13:45–15:25	Meeting
	14:08	11:30–14:40	Filling working hours
	14.08	14:40–15:00	Requirements analysis (coordination, references)
	15.08	12:15–13:05	Meeting
Sum	5	13:35	
W06	18.08	12:20–15:00	Requirements analysis (coordination)
	18.08	16:00–17:00	Meeting
	18.08	22:00–01:00	Requirements analysis (coordination)
	19.08	15:00–18:00	Requirements coordination, hardware requirements
	20.08	13:15–13:50	Meeting
	20.08	16:00–18:00	Software and hardware requirements
	20.08	18:00–18:15	Filling working hours
	22.08	23:00–23:35	Making personal page
	23.08	17:00–17:30	Meeting
	24.08	17:35–18:15	Use-cases: writing textual description
	24.08	18:15–18:40	Filling and checking working hours
	24.08	21:30–21:55	Making presentation slides
	24.08	22:00–00:10	Preparing for presentation
Sum	6	17:15	
W07	25.08	00:10–01:15	Use-cases: writing textual description
	25.08	13:00–13:15	Meeting
	25.08	13:15–17:00	Making diagram
	25.08	17:00–18:00	Making presentation slides

	25.08	20:00–21:00	Updating use-cases
	27.08	09:00–11:00	Presentation
Sum	2	09:05	
W09	08.09	12:00–13:00	Meeting
	10.09	23:00–02:00	UI design
	12.09	12:00–13:00	Meeting
Sum	3	05:00	
W10	15.09	12:00–12:40	Meeting
	16.09	22:00–23:05	Filling and calculating working hours
	17.09	09:50–13:00	Meeting, UI design
	18.09	09:45–13:00	UI design
	19.09	09:45–13:00	UI design: forms
	20.09	09:45–13:20	Searching for design tools, forms design
Sum	6	15:00	
W11	22.09	01:00–03:00	UI design: forms
	22.09	09:45–10:10	Filling woking hours
	22.09	10:10–13:00	UI design
	23.09	22:30–23:00	UI design: forms corrections
	24.09	09:50–13:00	UI design: forms, IO formats
	25.09	09:45–10:10	Filling working hours
	27.09	09:45–13:00	UI design: style, formats
Sum	5	12:35	
W12	29.09	10:00–13:00	UI design: formats
	01.09	09:45–10:15	Fillig working hours
	01.09	10:15–12:55	UI design: style convention, implementation notes
	01.09	12:55–13:05	Updating working hours
	02.09	09:45–12:05	UI Design: discussion
	02.09	12:00–12:30	Meeting
	02.09	12:30–13:05	UI Design: fixes
	03.09	10:00–13:05	UI Design: various fixes
	03.09	22:00–23:00	UI Design: forms fixes
	04.09	10:00–13:40	UI Design: various fixes
	04.09	13:40–13:50	Updating working hours
Sum	5	17:25	

W13	06.10	00:00–00:30	UI design: forms fixes
	06.10	09:50–13:00	UI design
	08.10	09:50–13:00	UI design
	09.10	09:50–13:05	Subsystems design: discussion
	09.10	22:30–01:00	Reading documentation
	10.10	09:45–10:10	Updating working hours
	10.10	10:10–13:05	Working on the UI document
Sum	4	16:05	
W14	13.10	09:45–11:00	Writing the UI document introduction
	13.10	11:00–13:05	Designing the Web-Server subsystem
	14.10	09:50–13:05	Designing the Web-Server subsystem
	15.10	09:50–13:05	Designing the Web-Server subsystem
	16.10	09:50–13:05	Designing the Web-Server subsystem
	17.10	09:50–13:05	Designing the Web-Server subsystem
	18.10	09:45–11:30	Writing UI style conventions
	18.10	11:30–13:05	Designing the Web-Server subsystem
Sum	6	19:40	
W15	20.10	09:45–13:05	Writing the formats specifications
	22.10	09:50–13:05	Designing the Web-Server subsystem
	23.10	09:50–13:15	Updating UI design documents
	24.10	09:50–13:05	Designing the Web-Server subsystem
	25.10	09:50–13:05	Updating forms
Sum	5	16:30	
W16	27.10	09:50–13:05	Writing unit testing specification
	29.10	09:45–13:10	Writing the Configuration and Installation document
	30.10	09:50–13:10	Writing the Configuration and Installation document
	30.10	19:00–21:00	Updating forms
	31.10	09:50–13:20	Making the initial source code structure
	01.10	09:50–13:25	Editing the build file and the configuration document
Sum	5	15:30	
W17	03.11	09:50–13:15	Editing the build file and the configuration document
	05.11	09:55–13:25	Starting the Web-Server implementation: the main page and the application deployment descriptor

	06.11	09:50–13:20	Implementing the main page and updating the deployment file
	07.11	09:45–13:10	Implementing the main page and updating the deployment file
	08.11	09:50–12:00	Implementing the main page and updating the deployment file
	08.11	12:00–13:20	Implementing the Session Manager
	09.11	14:00–16:15	Implementing the Session Manager
Sum	6	19:35	
W18	10.11	09:50–13:05	Implementing the Session Manager
	12.11	09:50–13:05	Implementing the Session Manager, creating the initial user profiles
	13.11	09:50–13:10	Implementing the Session Manager, fixing the Management and Data Store subsystems
	14.11	09:50–12:00	Fixing the Management and Data Store subsystems (for integration)
	14.11	12:00–13:10	Updating the web page design, implemenging the Log-In form
	15.11	11:30–12:50	Updating the web page design
	15.11	12:50–14:30	Implementing the Log-In form, Login servlet and the Registration form
	15.11	14:30–15:00	Updating user profiles
Sum	5	13:25	
W19	17.11	09:50–13:05	Unit testing
	19.11	09:50–13:20	Forms implementation and web-design
	20.11	09:50–13:10	Forms implementation and web-design
	21.11	09:50–13:15	Forms implementation and web-design
	22.11	11:40–15:10	Forms implementation and web-design
Sum	5	17:00	
W20	26.11	09:45–13:05	Forms implementation and web-design
	27.11	09:45–13:05	Forms implementation and web-design
	28.11	11:35–13:05	Forms implementation and web-design
	29.11	11:35–15:10	Forms implementation and web-design
Sum	4	11:45	
W21	01.12	01:30–02:10	Making new CS logo
	01.12	10:00–10:40	Meeting
	01.12	10:40–13:15	Implementing error handling
	01.12	14:30–16:40	Implementing error handling

	01.12	17:00–18:35	Implementing the notification page
	02.12	20:00–20:30	Implementing report and notification pages
	02.12	20:30–21:30	Web server: minor improvements
	02.12	21:30–22:30	Improving user profile locking mechanism
	03.12	10:30–13:15	Implementing the system set file uploading
	03.12	21:00–21:30	Implementing multipart/form-data reading for the Process servlet
	04.12	09:45–13:05	Web Server - Management integration: filling test plan
	04.12	19:00–20:30	Securuty fixes
	05.12	09:55–11:55	Fixing character encoding problems
	05.12	11:55–13:05	Web Server - Management integration: limits management
	06.12	11:30–15:00	Web Server - Management integration: limits and user management
Sum	6	23:25	
W22	08.12	09:55–13:05	Web Server - Management integration: limits and user management
	09.12	19:00–20:00	Web Server: improving error handling
	09.12	21:20–22:30	Web Server - Management integration: user management
	10.12	11:45–13:20	Web Server - Management integration: user management
	11.12	09:55–13:05	Web Server - Management integration: testing
Sum	4	10:15	
W23	15.12	09:55–13:05	Web Server - Management integration: testing
	16.12	21:00–23:00	Web pages design improvements.
	17.12	09:55–13:05	Web Server - Management integration: testing and fixing
	18.12	09:55–10:30	Web Server: fixing bugs
	18:12	10:30–11:25	Implementing the start page
	18.12	12:05–13:05	Testing and fixing
	19.12	09:55–11:30	Web Server - Activity Statiscs integration: fixing the “null IP address” bug
	19.12	11:30–13:05	Testing and fixing Algorithm Server initialization problems
	20.12	11:30–13:30	Meeting, improving the Web Server
Sum	6	16:00	
W24	23.12	23:00–00:00	Web Server: documentation and forms updates
	24.12	20:00–21:00	Editing the website header

	25.12	23:00–01:20	Making new website header
Sum	3	04:20	
W27	16.01	07:20–07:55	Fixing the prompt page problem
	16.01	08:00–10:15	Implementing the “save ANLDE System (Set)” function
	16.01	12:50–13:05	Testing and improoving the Web-Server subsystem
	16.01	15:50–18:45	Testing and improoving the Web-Server subsystem
	18.01	00:30–01:30	Testing and fixing the report page
Sum	2	07:05	
W29	28.01	16:00–19:50	Testing and improoving the Web-Server subsystem
	29.01	17:00–17:45	Testing and improoving the Web-Server subsystem
	29.01	20:00–21:05	Testing and improoving the Web-Server subsystem
Sum	2	05:40	
W30	02.02	16:10–16:30	Meeting
	02.02	17:00–19:00	Analysing and filling working hours based on CVS history
	02.02	20:00–20:30	Analysing and filling working hours
	04.02	16:00–17:00	Analysing and filling working hours
	05.02	15:00–15:45	Analysing and filling working hours
	05.02	16:10–16:20	Analysing and filling working hours
	05.02	16:20–16:50	Calculating working hours
	05.02	17:00–17:25	Analysing and filling working hours
	05.02	17:25–17:40	Calculating working hours
	06.02	16:00–17:00	Updating the personal page
	06.02	17:00–19:00	Testing and fixing: report pages
Sum	4	08:55	
W32	17.02	21:00–00:00	Testing and fixing: report pages and algorithm server
Sum	1	03:00	
W33	25.02	17:00–18:30	Writing user manual: installation and configuration
Sum	1	01:30	

Month	**Time spent (dd; hh:mm)**	**Process phase**
July	9; 14:55	Requirement analysis
August	20; 61:30	Requirement analysis
September	15; 35:35	Design
October	23; 82:10	Design
November	21; 68:35	Implementation
December	19; 54:00	Implementation, testing
January	4; 12:45	Alpha-testing
Febrary	6+; 12:25+	Alpha-testing
Total	117+; 344:55+	

7 List of Main Artifacts

1. **Maintenance document.** This one.
    - General, see section 1;
    - Project Team, see section 2;
    - Problem Domain, see section 3;
    - User Requirements, see section 4;
    - Project Plan, see section 5.
2. **Detailed Meetings Chronology**.
3. **Development notes:**
    - Requirements Specification;
    - Design;
    - Test plan;
    - Implementation document;
    - Test execution document;
4. **User Guide.**
5. **Source code. Not free available.**
6. **The Web-SynDic System**
7. **Final report. Under construction.**

8 List of chosen tools

You can find the list and downloads here.

References

[1] Miguel Filgueiras M., Ana-Paula Tomás. *Solving Linear Constraints on Finite Domains through Parsing.* In P. Barahona, L. Moniz Pereira, A. Porto (eds.), Proceedings of the 5th Portuguese Conference on Artificial Intelligence, Springer-Verlag, 1991. LNAI 541, pp. 1–16.

[2] Dmitry G. Korzun. *Syntactic Algorithms for Solving Nonnegative Linear Diophantine Equations and their Application for Modelling of Internet Link Workload Structure.* PhD Thesis, Department of Computer Science, University of Petrozavodsk, 2002. 185 p.

[3] Dmitry G. Korzun. *Grammar-Based Algorithms for Solving Certain Classes of Nonnegative Linear Diophantine Systems.* Proceedings of Annual international Finnish Data Processing Week at the University of Petrozavodsk (FDPW'2000): Advances in Methods of Modern Information Technology. Vol. 3. Petrozavodsk, 2001, pp. 52–67.

[4] Schrijver A. *Theory of linear and integer programming.* Wiley, Chichester, 1986.

[5] Huet G. *An algorithm to generate the basis of solutions to homogenous linear diophantine equations* // Information Processing Letters, 1978. Vol. 3. No. 7. PP. 144–147.

[6] Domenjoud E. *Solving Systems of Linear Diophantine Equations: An Algebraic Approach.* In U. Tarlecki (ed.), Proceedings of 16th International Simposium on Mathematical Foundations of Computer Science. Springer–Verlag, 1991. LNCS 520. PP. 141–150.

[7] Roger S. Pressman. *Software Engineering. A Practitioner's Approach.* European adapt., 5th ed. McGraw-Hill, 2000. 915 p. ISBN 0-07-709677-0.

[8] Ian Sommerville. *Software Engineering.* 6th ed. Addison-Wesley, 2000. ISBN 0-201-39815-X.

[9] Graig Larman. *Applying UML and Patterns.* Prentice Hall, 2000. ISBN 0-13-748880-7.

[10] Jim Conallen. *Building Web Applications with UML.* Addison-Wesley, 2000. ISBN 0-201-61577-0.