Proceedings of the 9th International Conference "Reliability and Statistics in Transportation and Communication" (RelStat'09), 21–24 October 2009, Riga, Latvia, p. 404-412. ISBN 978-9984-818-21-4
Transport and Telecommunication Institute, Lomonosova 1, LV-1019, Riga, Latvia

APPLICATION OF THE AGENT-BASED SIMULATION APPROACH TO THE BANK DEPARTMENT FUNCTIONING ANALYSIS

Elena Yurshevich, Aleksandra Kalna, Andris Kalns

Transport and Telecommunication Institute
Lomonosova Str. 1, Riga, LV-1019, Latvia
Ph.: +371 67100651. Fax: +371 67100660. E-mail: elena_y@tsi.lv
Ph.: +371 26455558. E-mail: alek_sandra@inbox.lv
Ph.: +371 26114514. E-mail: andri@navigators.lv

This paper **deals with** the results of agent-based simulation approach application in the field of process management. **T**he problem of the bank to bank payment department efficiency increasing with the aim the annual budget minimisation **has been considered**. The main bank department responsibility to process and investigate bank to bank payments using SWIFT telecommunication network, advise customers via phone and e-mail. During the workflow investigation some hypothesis has been proposed that the present logic of work organisation causes inefficient distribution of the resources and as result high expenses and time delays. The last one is very crucial index of payments processing efficiency. It was decided to implement the model of the department functioning using the **agent-based** approach. **S**everal types of agents: operators, SWIFT messages, phone calls and e-mails **have been implemented**. They interact and influence each other and as a result the model reproduces the logic of real system functioning. The model **has been** implemented on the base of simulation **modelling** tool AnyLogic, which supports the **agent-based** approach. The results of experimentation allow defin**ing** different options of working process organization, which give the capability to reduce expenses and to improve efficiency of work.

Keywords: *system analyses, agent-based simulation modelling, business process organisation, bank to bank transfers, budget, process efficiency, experimentation,* ***analyses***

1. Introduction

Under the conditions of deterioration of the economic conjuncture for enterprises financial stability maintenance the primary targets are business processes optimisation and efficiency increasing. One of the extremely effective ways of the system investigation is the simulation modelling. It is one of the powerful tools of the system behaviour analysis and its optimisation. The simulation can be used at different stages of the systems life-cycle development, which includes planning, analysis, design, implementation, maintenance and reengineering. The main aim of the simulation application is the risk minimisation and finding of the best way of the idea implementation in the practice.

Depending on the level of the system description abstraction there can be used different approaches to simulation: system dynamic, discrete-event simulation and agent-based simulation [1]. In its turn the approach selection depends on the aim of simulation application, the level of decision-making and the system's complexity. Considering the system's dynamics it is frequently used for analysis of very complex large systems (for example, the social systems, country, regions or urban economics, etc). The size and complexity of such systems require the description of the process as continuous one, with high level of abstraction [1]. The discrete-event simulation is used for the modelling of the processes at the middle level of their particularity description. As a rule the components of such processes change their states at the discrete time moments and the dynamics is represented by the dynamic entities appearing. Such approach is used for description of the logistics, productions and others systems. The agent-based approach is the modern approach for the complex system investigation with the high level of its particularity. The agent-based approach is used commonly for the exploring the system which behaviour is a result of different types of dynamic entities interactions [1, 2]. The multi agent metaphor, perceiving a dynamic system as a community of autonomously interacting entities, has started permitting many application areas of modelling and simulation [1].

This paper describes the application of the agent-based approach to the investigation of the bank department functioning. The main bank department responsibility is to process and to investigate the bank to bank payments using SWIFT telecommunication network and advise customers via phone and e-mail. The efficiency of this department functioning depends on the efficiency of the process organization logics on the whole, and the efficiency of staff interactions, in particular. That's why it was decided to choose the agent-based approach to this department functioning description.

2. Simulation Object Description and the Problem Definition

The considered bank department main responsibilities are the following: searching for EUR correspondent chain for customer's payments; performing and investigating financial institution transfers. This department arranges the daily communication with customers (banks, stakeholders, other branches and departments) via phone SWIFT network and e-mail, advising them about inquiries of the concerned transactions.

Efficiency is a ratio of the result divided into the consumed resources. The requirements for the service quality are pretty high; all incoming payment transfers must be booked before the cut-off time for certain currencies, providing execution of non STP payments during operational hours (Straight Through Process, payments suspended by the bank payment system). All inquires via SWIFT network must be handled within 24 hours; inquires received via e-mail and phone answered immediately. To increase the efficiency without reducing the quality of service level it is necessary to reduce expenses. The hypothesis proposes that the problems existing in the bank department originated due to not well organized business process, redundant execution of the tasks, not optimal distribution of work responsibilities that caused delays while handling investigations, error costs due to late execution. To solve these problems it has been proposed:

- to change the number of operators,
- to reorganize the business processes,
- to change the salary calculation scheme and to reduce the overtime hours that has to be paid with double fair.

The simulation modelling approach is chosen to test this hypothesis.

3. Conceptual Model Description

Before simulation model implementation conceptual model has been created, in order to find the possibilities of increasing efficiency of the working process analysing workflows. The conceptual model describing the workflow logics in the department is presented on Figure 1.

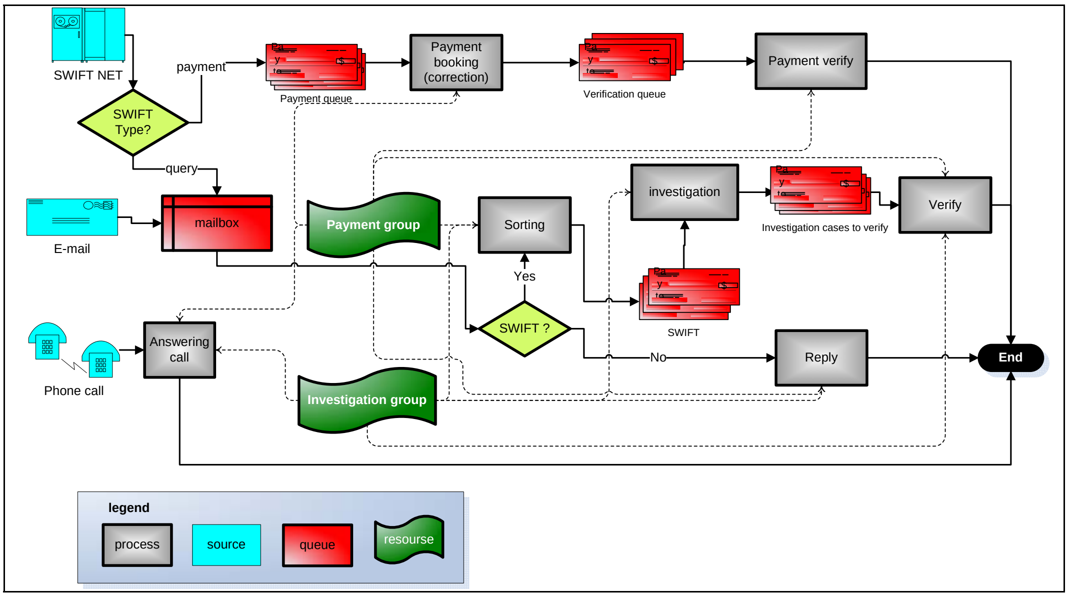

Figure 1. Conceptual model scheme of workflow in the bank department

In the department there are working 8 specialists, who are divided into investigation and payment groups. All incoming SWIFT messages are divided into payment instructions that go to payment system and inquiries (investigation requests) that are sorted automatically and routed to relevant department. Investigated SWIFT messages are printed out and sorted by specialists to several groups defined by request types.

Two specialists are assigned to payment group every day; their task is to book payments, that haven't passed through the system automatically and after the manual interruption another person should verify corrections done by the first specialist. Investigations are handled by the rest of the team, who investigates on customer's request already having been executed payments. Each specialist is handling the appropriate group – certain investigation type. The investigation of one case (payment) can last even for several weeks, because it requires the communication with other banks via SWIFT network. Replies and requests that have to be sent out via SWIFT network must be also verified by another specialist.

E-mails replying and SWIFT message sorting are handled by the whole team and responsibilities are divided according to the appropriate hours and received time. The received phone calls must be answered immediately within three ring tones that mean the interruption of task specialist currently on.

There are defined the external and internal parameters, which have the influence on the system behaviour. The most important parameters are as follows:

- external parameters:
  - time intervals of the phone calls appearing (min),
  - time intervals of the payment instructions received during day and night time separately (min),
  - time intervals of the e-mails appearing (min),
  - time intervals of the SWIFT message received interval from 9 till 6 p.m. with the time gap in 1 hour (min),
  - time intervals of the SWIFT message received interval during night time (min);
- internal parameters:
  - time delays to phone calls answers (min),
  - time delays to correct the payment (min),
  - time delays to verify the corrected payment (min),
  - time delays to answer e-mail (min),
  - time delays to sort SWIFT message (min),
  - time delays to handle investigation (min),
  - time delays to verify investigation case (min),
  - number of operators who process the calls, payments, e-mails, etc.

During working process supervision in the department the statistics of mentioned parameters has been collected. The data are processed in STATISTICA application, in order to use it on the stage of conceptual model formalization. The distribution functions of the model parameters are estimated with STATISTICA. The results are displayed in Table 1.

Table 1. External model parameters distribution functions

Model Parameter	**Distribution function and parameters estimations**
Phone calls received interval	Lognormal (min=0.67; μ^*=2.93;σ^*=0.86)
Payment instructions received interval during day time	Lognormal (min=0.46; μ^*=0.25;σ^*=0.22)
Payment instructions received interval during night time	Lognormal (min=5.23; μ^*=2.95;σ^*=0.25)
E-mails received interval	Lognormal (min=1; μ^*=2.41;σ^*=1.21)
SWIFT message received interval from 9 till 10 a.m.	Lognormal (min=1; μ^*=2.00;σ^*=1.38)
SWIFT message received interval from 10 till 11 a.m.	Lognormal (min=1; μ^*=1.85;σ^*=1.27)
SWIFT message received interval from 11 till 12 a.m.	Lognormal (min=1; μ^*=1.54;σ^*=1.43)
SWIFT message received interval from 12 a.m. till 1 p.m.	Lognormal (min=1; μ^*=1.78;σ^*=1.16)
SWIFT message received interval from 1 till 2 p.m.	Exponential (min=1; λ^*=0.13)
SWIFT message received interval from 2 till 3 p.m.	Lognormal (min=1; μ^*=1.52;σ^*=1.49)
SWIFT message received interval from 3 till 4 p.m.	Lognormal (min=1; μ^*=1.45;σ^*=1.29)
SWIFT message received interval from 4 till 5 p.m.	Lognormal (min=1; μ^*=1.22;σ^*=1.14)
SWIFT message received interval from 5 till 6 p.m.	Lognormal (min=1; μ^*=1.33;σ^*=1.16)
SWIFT message received interval during night time	Exponential (min=3; λ^*=0.02)

Table 2. Internal model parameters distribution functions

Model Parameter	**Distribution function and parameters estimations**
Time spent to answer phone call	Lognormal (min=0.17; μ^*=0.25;σ^*=0.94)
Time spent to correct the payment	Lognormal (min=0.03; μ^*= -1.36;σ^*=0.64)
Time spent to verify the corrected payment	Lognormal (min=0.17; μ^*= -1.90;σ^*=0.49)
Time spent to answer e-mail	Lognormal (min=0.6; μ^*=1.59;σ^*=0.86)
Time spent to sort SWIFT message	Chi-square (min=0.3; μ^*=2.78)
Time spent to handle investigation	Lognormal (min=1.8; μ^*=2.24;σ^*=0.48)
Time spent to verify investigation case	Exponential (min=0.2; λ^*=1.86)

Time delays to process tasks and received intervals are measured in minutes and due to dynamic and stochastic behaviour of the system some of the parameters differ depending on the daytime and season. Times delays to process tasks were collected during 2 months, but received intervals were taken by the data existing in mailbox and daily statistic reports collected in 2008.

4. Description of the Model Implementation

It has been decided to implement the bank department model using the agent-based approach, because of the fact that it is more attractive for the modelling tasks. Using this approach it is necessary to know only the low level logics, basically, how agents act, but the macro level logics appears as an agent interaction result. In fact, it is highly flexible and reacting rapidly on the system changes. This approach is based on the objects called "agent", it is some entity, which can be simulated by the behaviour model. Behaviour is expressed with the agent state changes and events. The AnyLogic 6 has been chosen as a simulation tool for the model implementation, because of the fact that it supports the agent-based approach. The simulation time unit is selected as 1 minute, which gives reasonable precision for model parameters and, on other hand, does not require too many unnecessary simulation iterations. The simulation period is 1 year. The agent's logics implementation is demonstrated by several examples of the implemented agents.

Analysing department workflow as a system, the following agents have been defined:

- Specialist – employees, who perform work tasks manually
- SWIFT messages
- E-mails
- Phone calls

All these agents have their own characteristics, essence and logic of behaviour. It is quite simple to define logic for none-live agents at the point similar to discrete-event experience. But modelling human behaviour in current situation in every particular moment becomes real programming challenge unless you model with the agent-based approach. Since all the objects interact and influence each other, having their own behaviour, and as a result, the agent-based simulation model reproduces the logics of real system functioning quite close to reality in particular aspects. The agent implementation in the model is based on the state charts, the set of rules, functions and procedures. During the model construction there have been implemented several state charts for each agent. Some of them are described in the paper. The example of the agent "SWIFT message" state chart is presented on Figure 2.

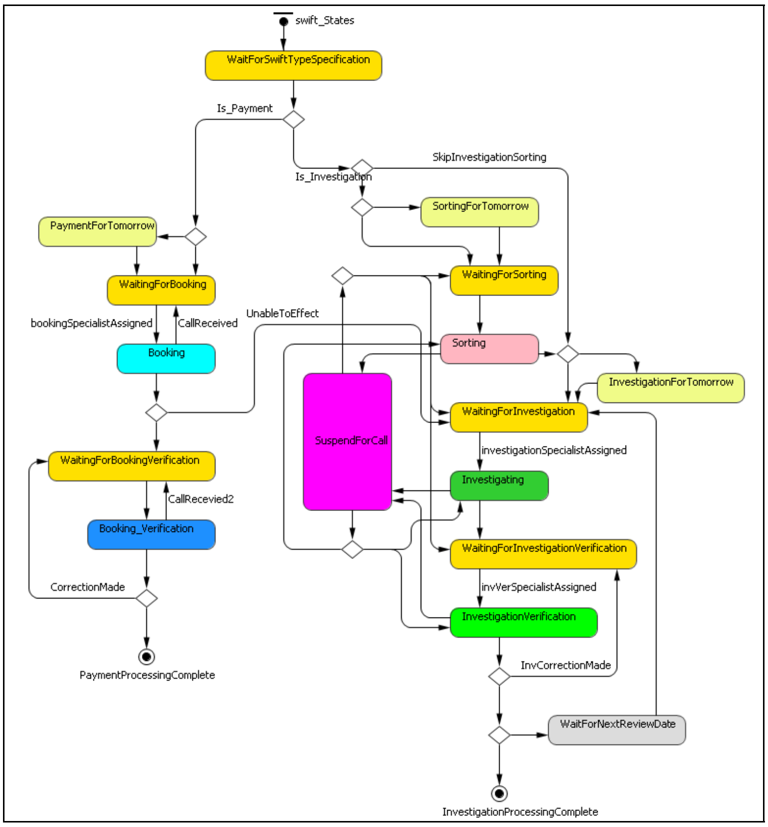

Figure 2. State chart of the agent "SWIFT message"

The message arriving at the system does not yet know if there is the payment instruction or the inquiry about another payment. Once type is clarified it changes the state based on its type and arrival period. Payments or inquiry, which arrive too late for today, are put for to-morrow queue and won't be considered as a task for today. The item's state changes started together with its processing. If after processing the item must pass verification and the required resource (specialist) is not available, the item will wait for another waiting queue (waiting state). State change logics represents real system logics, but it is defined in agent-based model on the low detail level.

Another example of state chart for agent is presented on Figure 3. The agent called "Specialist" in the model may have different states. The simplest states are as follows:

- AtHome,
- OnVacation,
- AtWork.

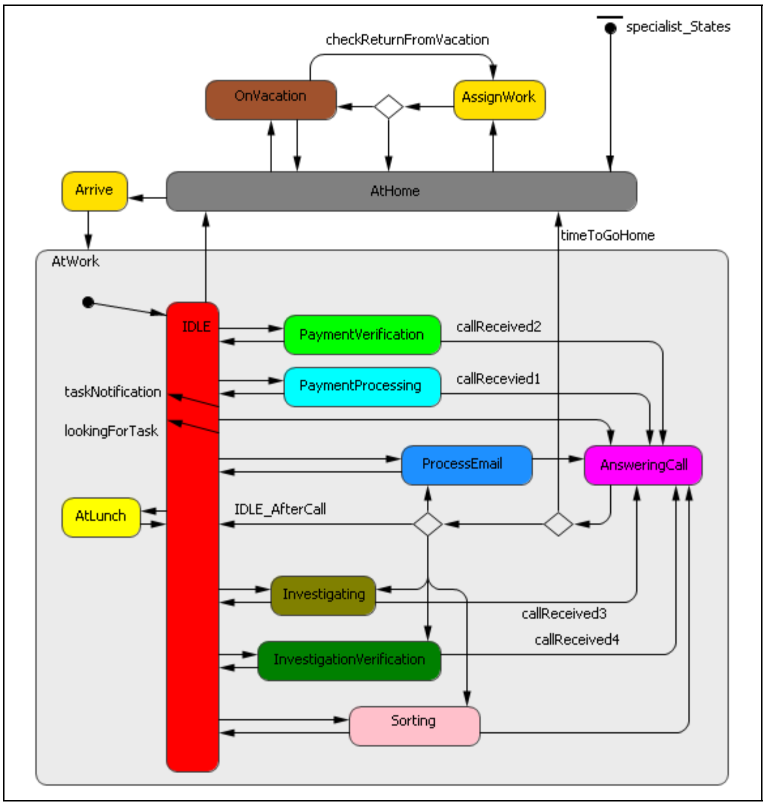

Figure 3. State chart atWork of the agent **"**Specialist"

For the additional logics modelling the utility states have been created:

- Arrive – to simulate random arrival time due to random travel time,
- AssignWork – given total expected specialist count for coming day assign duties, optionally recalled from vacation.

However, we need to know more details about the specialist currently atWork. In model it is defined as the complex state, which has nested states, the specifying action employee performs.

Like it happens in real life, specialist is altering items (emails, phone calls, SWIFT message) by processing them, in model this is done by controlling state changes of agents who present the work items (emails, swifts, calls).

Every agent "Specialist" in the model can perform any available task type, but decision to do so is fully controlled by the agent's logics. Each decision is made individually and can take into account any model runtime parameter and model statistics. For example:

- does he need to help colleague, if it's clear, that it's not possible for colleague to complete work today?
- are there items "for tomorrow" he can process today?
- are there items, which he can verify himself – he is not verifying his own items?

- will specialist react on call for help from colleague?
- will he take task from queue or stay in "idle" state, just to rest?
- can specialist go to lunch or on vacation at this particular moment?
- does he suspend current task if the call is received?

On every state transition in model it is possible to perform the additional task, this way specialist can even try to search for new task while currently idling without work.

The other agents have been implemented in the similar manner. On the whole the model contains 4 state charts, 73 functions, 4 agents, 1 active object, 7 utility classes and 21 scenarios of the simulation experiments.

During the implementation of the simulation model it is verified using test run methods, to be sure the model has been implemented with correct algorithms and used certain data in certain moment of time. Before experiments executions model set with all real life parameters is configured. It has been validated using the confidence intervals methods which demonstrate satisfactory results.

Once base model is created, verified and validated against real department statistics, experiments have been designed and executed.

5. Experiments Implementation

For experiments implementations there have been prepared, in general, 47 model parameters, which are possible to set-up externally in the experiment definition. However, based on analysis of department work as a system, only 8 parameters are changed to prepare combinations of values forming experiment set-up.

Test runs of the model implemented in AnyLogic 6.4 environment on the base of quad-core Intel CPU (Q6600) for 1 year, showed that elapse time was around 60 seconds. The experiment results were stored in external database. This give the opportunity to accumulate several experiments runs. Although 1 simulation run itself is technically hard to make it parallel to different threads to be executed on separate CPUs/Cores, it is rather easy to start programmatically multiple experiments in parallel and let them run on different cores. This way, by use of database as storage for experiment results, model has not been limited to single thread and it is possible to implement massive execution of experiments, spread to different CPU cores or even to remote machines which will run their portion of experiments.

The efficiency criteria of department work was the ratio of "quality+quantity" of service divided by expenses, therefore the main aim was to decrease budget of the department, while maintaining the same service level and processing volumes.

To ensure requirements for simulation goal there have been implemented the following output characteristics:

- department total year and monthly budget (costs and expenses),
- daily reports on specialist utilization, including utilization distribution between team members,
- daily reports on overtime hour count,
- daily reports on complete/incomplete/waiting in queue items,
- daily histogram on time spent in department until processed for payments and inquiries.

The main parameters influencing the department workflow entered the simulation model as individual experiments. Experiments were designed to check the following cost cutting ideas:

- personnel decrease – most obvious but uncertain result in long term,
- business process reorganization – elimination of unimportant tasks,
- personnel increase – in attempt to eliminate overtime pay expenses,
- forbid overtime – cancel overtime duty and logically overtime pay,
- employ more professionals – faster processing, less errors, less expenses,
- employ only middle qualification personnel – leveraged performance and expense rates,
- employ only junior level specialist – pay less, hire more, fire once becomes too expensive specialist,
- simulate automated workflow system, which would divide workload honestly and support real time workload management, including task reassignment, alerts etc.,
- combinations of previous items taking into account theoretical influence on results.

When the first results had been received, it was clear, that personnel utilization was unevenly distributed between team members, forcing some of them to take overtime hours, while others idled during the day for free. For that reason additional experiments were added in attempt to make individual workload more even and decrease overtime:

- reorganize duties during the day to different groups inside the team,
- assuming there are less than 12 people in team, limit count on vacation is 1 or 2 person.

For experiments several parameters were changed:

- personnel number;
- swift message sorting task (active/cancelled);
- overtime hours allowed (yes/no);
- limit on vacation count (1/2/3);
- automated workflow system (yes/no);
- personnel qualification (number of skilled/novice/professionals);
- specific task assignments (groups inside the team).

There have been prepared 20 experiments, which are presented in Table 3.

Table 3. Experiment's scenarios

Sce-nario	Wor-kers count	SWIFT messa-ge sorting	Salary calcu-lation sheme	Over-time allowe d	On vaca-tion count	Automat ed system	Qualification (expert, specialist, beginner)	e-mails are handled by
base	8	yes	fixed	yes	2	no	2:5:1	all
1	7	-	-	-	-	-	2:4:1	-
2	-	no	-	-	-	-	-	-
3	7	no	-	-	-	-	2:4:1	-
4	9	no	-	-	-	-	2:6:1	-
5	-	no	-	no	-	-	-	-
6	-	no	-	-	-	-	4:4:0	-
7	-	no	-	-	-	-	8:0:0	-
8	-	no	-	-	-	-	0:4:4	-
9	-	no	-	-	-	yes	-	-
10	-	no	-	-	-	-	-	payment gr.
11	7	no	-	-	-	-	2:4:1	payment gr.
12	7	no	-	-	1	yes	2:4:1	-
13	-	no	-	no	1	yes	-	-
14	-	-	-	-	1	-	-	-
15	-	no	-	-	1	-	-	payment gr.
16	7	no	-	-	1	-	2:4:1	payment gr.
17	-	-	not fixed	-	-	-	-	-
18	7	no	not fixed	-	-	-	2:4:1	-
19	7	-	not fixed	-	-	-	2:4:1	-
20	-	no	not fixed	-	-	-	-	-

The Table 3 shows the alternative scenarios parameters that have been stated in the model, dash (-) means parameters don't differ from the Base model configuration, which is set up as a real life department.

Each experiment has been conducted for 50 times with the random seed on probability generators. The results after experimentation have been aggregated and analysed in order to choose the efficient alternative scenarios.

6. Experimentation Results Analysis

Some results (the mean value of the department yearly expenses) of the experimentation with the model are presented on Figure 4. Evidently all the experiments having higher costs than in the base model do not satisfy our demands, even if the quality of service is higher. Also, the rest scenarios with smaller expenses might not be so efficient, since work tasks would not be done on time. Therefore before conclusions all daily statistics of accomplished and not-accomplished in time tasks of implemented experiments has been analysed as well. The experiments demonstrated that the hypothesis that the salary calculation scheme will minimize costs, failed.

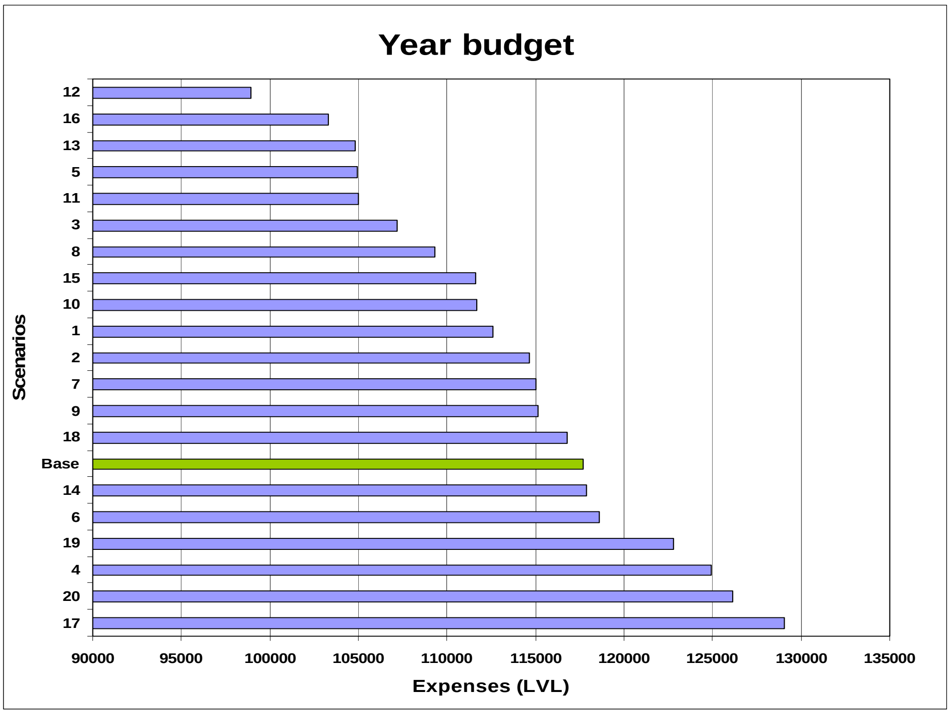

Figure 4. Experiment results

For fewer analysis there were chosen only the scenarios, which expenses were not higher than in the base model and service agreements carried out.

Also after choosing more satisfactory scenarios, additional experiments have been implemented to see what consequences will be if one more person is out of the office or due to the unexpected amount of work.

7. Conclusions

The application of the agent-based modelling approach allowed testing the proposed hypotheses of the bank department efficiency improving ways. The implemented simulation model results demonstrated that the most efficient scenario is the 12th when:

- 7 persons are employed;
- at the same time, only one specialist is on vacation;
- print outs and paper-based cases are avoided, electronic equivalents are used;
- cancelled SWIFT message sorting, performing investigations without dividing into groups;
- automatic workflow system is used.

This scenario gives the possibility to decrease annual expenses for 16%. However, the implementation cost of the automated workflow system was not included into the scope of this work. We can assume that this solution will require the integration with the other IT systems in the bank and business process alignment. Such solution can be accepted only if it suites several if not all the departments. Even if this system operates with generic tasks and their relative "costs", business requirements consolidation between different parties would require significant time and effort.

Taking into account such unlikely quick-cheap solution for the automated workflow system, the following most efficient scenario is similar to the 12th, but instead of the automated system additionally there are done responsibility reassignments to groups inside the team. In this case (scenario 16) – year expenses can be cut for 12% (118-103 thousand LVL).

The implemented model can be used as a tool for analysing alternatives of business process organisation. That gives a possibility to analyse weak places and find solutions how to make the system – real bank department functioning in more efficient way prior to real changes. After all, this simulation model empowers the executives with the realistic figures before making decision.

References

1. *Handbook of Dynamic System Modelling/ Edited by Paul A. Fishwick*. University of Florida, Gainesville, USA: Chapman & Hall/CRC, 2007, pp. 8-1 – 8-12.
2. Borshchev, A. Filippov. From System Dynamics and Discrete Event to Practical Agent Based Modeling: Reasons, Techniques, Tools. In: *Proceedings of the 22nd International Conference of the System Dynamics Society.* Oxford, England, 2004,
3. Garifullin, M., Borshchev, A., Popkov, T. Using AnyLogic and agent-based approach to model consumer market. *EUROSIM 2007, September 9-13 2007, Ljubljana, Slovenia.*
4. Application of agent-based modelling – http://www.xjtek.com/anylogic/approaches/agentbased/