RESEARCH IN LOGISTICS & PRODUCTION

LOGISTICS OF SANITARY TEAMS ACTIVITIES

Radosław Pytlak*, Damian Suski*, Tomasz Zawadzki*
and Wojciech Stecz**

* Institute of Automatic Control and Robotics, Warsaw University of Technology,
Sw. Andrzeja Boboli, 02-525 Warsaw, Poland, Email: r.pytlak@mchtr.pw.edu.pl
** Faculty of Cybernetics Military University of Technology, Kaliskiego 2,
00-908 Warsaw, Poland, Email: wstecz@wat.edu.pl

Abstract We describe the information system that has been built for the support sanitary teams. The system is aimed at supporting analytical work which must be carried out when there is a risk of epidemic outbreak. It is meant to provide tools for predicting the size of an epidemic on the basis of the actual data collected during its course. Since sanitary teams try to control the size of the epidemics such a tool must model also sanitary teams activities. As a result a model for the prediction can be quite complicated in terms of the number of equations it contains. Furthermore, since a model is based on several parameters there must be a tool for finding these parameters on the basis on the actual data corresponding to the epidemic evolution. The paper describes the proposition of such a system. It presents, in some details, the main components of the system. In particular, the environment for building complex models (containing not only the epidemic model but also activities of sanitary teams trying to inhibit the epidemic) is discussed. Then, the module for a model calibration is presented. The module is a part of server for solving optimal control problems and can be accessed via Internet. Finally, we show how optimal control problems can be constructed with the aim of the efficient epidemic management. Some optimal control problems related to that issue are discussed and numerical results of its solution are presented.

Paper type: Research Paper

Published online: 31 July 2015
Vol. 5, No. 4, pp. 393-406

ISSN 2083-4942 (Print)
ISSN 2083-4950 (Online)

Keywords: *Epidemic models, Forrester's models of sanitary activities, Model calibration, Optimal management of sanitary activities*

1. INTRODUCTION

The discussed environment for supporting activities of sanitary teams consists of several components: database, modelling and simulation component and optimization component.

The main component is for modelling the epidemic evolution. Furthermore, the modelling component must have the ability for presenting activities of sanitary teams since these activities influence the spread of epidemic. The component uses graphical interface to facilitate building models of complex systems – the modelling component is discussed in the next section.

The other important component of the system is the component that is responsible for performing several tasks related to the optimization of a dynamic model which is provided by the modelling component. It can carry out tasks such as: calibration of the model; optimization of sanitary teams activities aimed at reducing the spread of epidemic. The optimization component is in fact IDOS server which accessed by Internet.

Besides these two main components our environment has also components which provide necessary data for the main components. There is a database (DB component) which stores information on the current number of people inflicted by diseases, as well as historical data related to previous epidemics. It contains also data which inform about food storage and its consumption – in particular these data are needed to model the evolution of food borne diseases.

Furthermore there are also components: DOML File Generator and Optimizer GUI. DOML File Generator is needed to transform data from the modelling component to data suitable for IDOS Optimizer. As is mentioned in section on the optimization component the transformation is in fact rewriting of differential-algebraic equations as ordinary differential equations. Optimizer GUI enables defining the optimization problem by inserting into a DOML file statements such as objective function and various kinds of constraints.

2. MODELLING COMPONENT OF THE SYSTEM

The modelling component uses the Forrester's methodology for building dynamical models of complex systems (Sterman, 2000). The methodology was pro-

posed by Forrester in late fifties of the previous century and initially was applied to describe dynamical behaviour of activities of any organization, in particular industrial company. Later the methodology was successfully applied to any kind of human activities including decision making process.

According to the Forrester's methodology the model of a process is built in two stages. In the first stage main variables of the process are listed and the relations between them are stated. In that stage so-called Casual Loop Diagram (CLD) is constructed which reveals closed loops in the process which can be analyzed and used for the verification of the correctness of the process model. For example, if the process has variables which historical values show exponential growth, there must be at least one positive closed loop in the CLD of the process.

In the second stage the CLD of the process is transformed into a set of differential-algebraic equations. It is done by stating that some variables in the CLD must be differential variables which means that there must be differential equations associated with these variables, the other variables of the CLD define algebraic equations. In the second stage to each variable (differential, or algebraic) some function (in general nonlinear) is assigned. At the end of the modelling process we have a set of differential-algebraic equations.

In Fig. 1 we present an example of the model built with the help of the Forrester's methodology and the application Vensim.

The model presented in Figure 1 has several components. The first component describes the process of generating reports on cases of disease. The reports must be verified in order to find whether the cases are related to the same disease and whether the number of cases suggests the epidemic.

The part of the model corresponding to the extension of the classical SIR model of epidemic (see (Anderson, May, 1991), (Bailey, 1975), (Kermack, McKendrick, 1927), (Kermack, McKendrick, 1932), (Kermack, McKendrick, 1933), (Murray, 1993), (Murray, 2001)) - the submodel describes the spread of food borne diseases - is given in Fig. 3. Squares boxes in Figure 2 correspond to differential variables while circles to algebraic variables. In the submodel we have two variables Inf_Popul_A and Inf_Popul_B which correspond to populations of infected people. We assume that people from one group may be infected by contact with infected people and people from the other group by contact with infected people, or by contact with contaminated food.

The integrated model of sanitary inspections teams activities includes the model of decision making. We assume that sanitary teams can curb the spread of epidemic by sending enough sanitary inspections teams to people suspected to be ill (in that way these people could be isolated), or by sending enough sanitary inspections teams to places with contaminated food (in that way contaminated food could be destroyed). The analysed process contains several decision rules, all of them link creation of sanitary inspections teams with available funds. The decision rules are used in the definition of variables: rate of new inspections, rate of dropping food inspections, rate of new infected people inspections, rate of dropping people inspections.

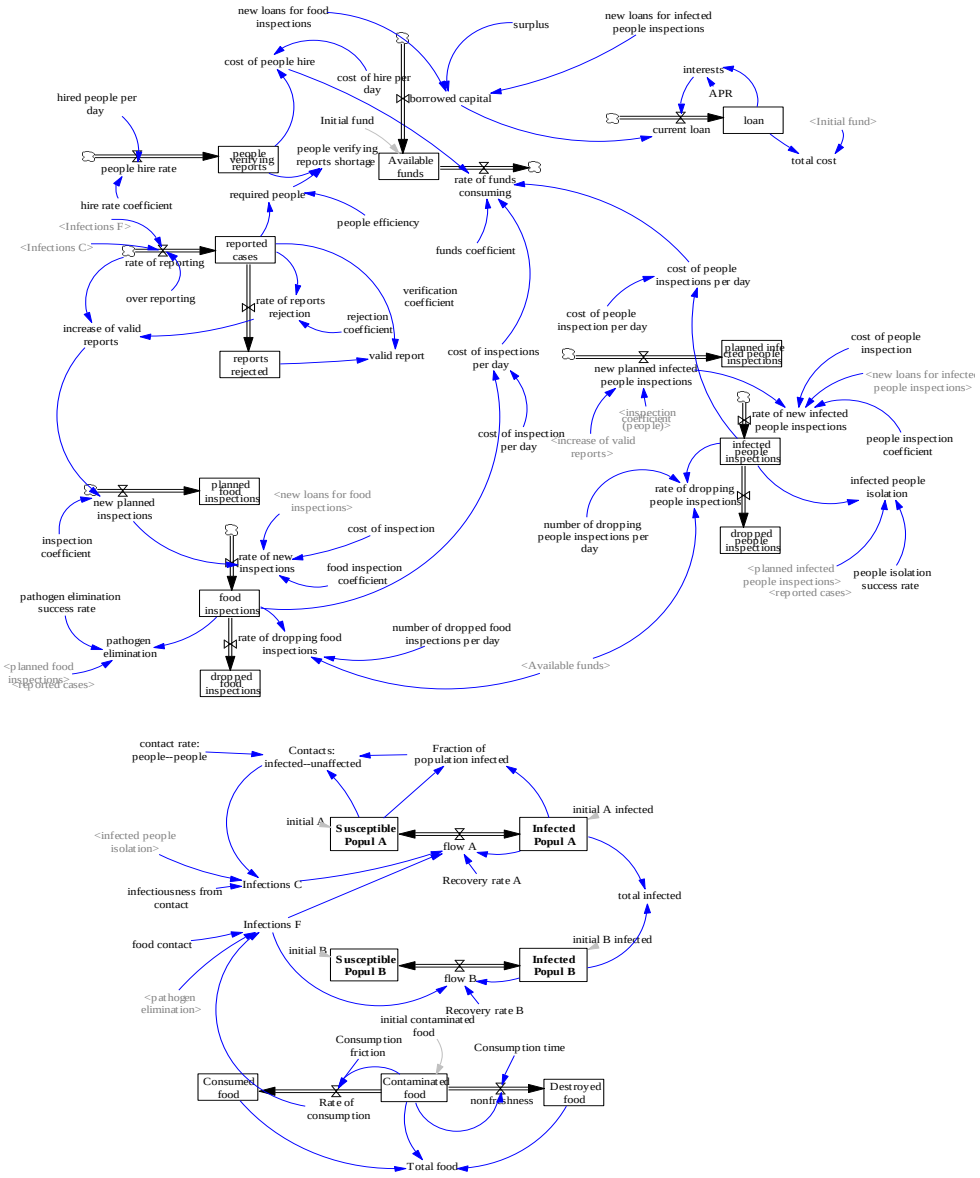

Fig. 1 Integrated model of sanitary inspections teams activities and the extended SIR model

Decision rules associated with these variables have similar constructions, therefore we analyse in detail the decision rule which set the value of variable rate of new infected people inspections. The formula for the variable rate of new infected people inspections is as follows:

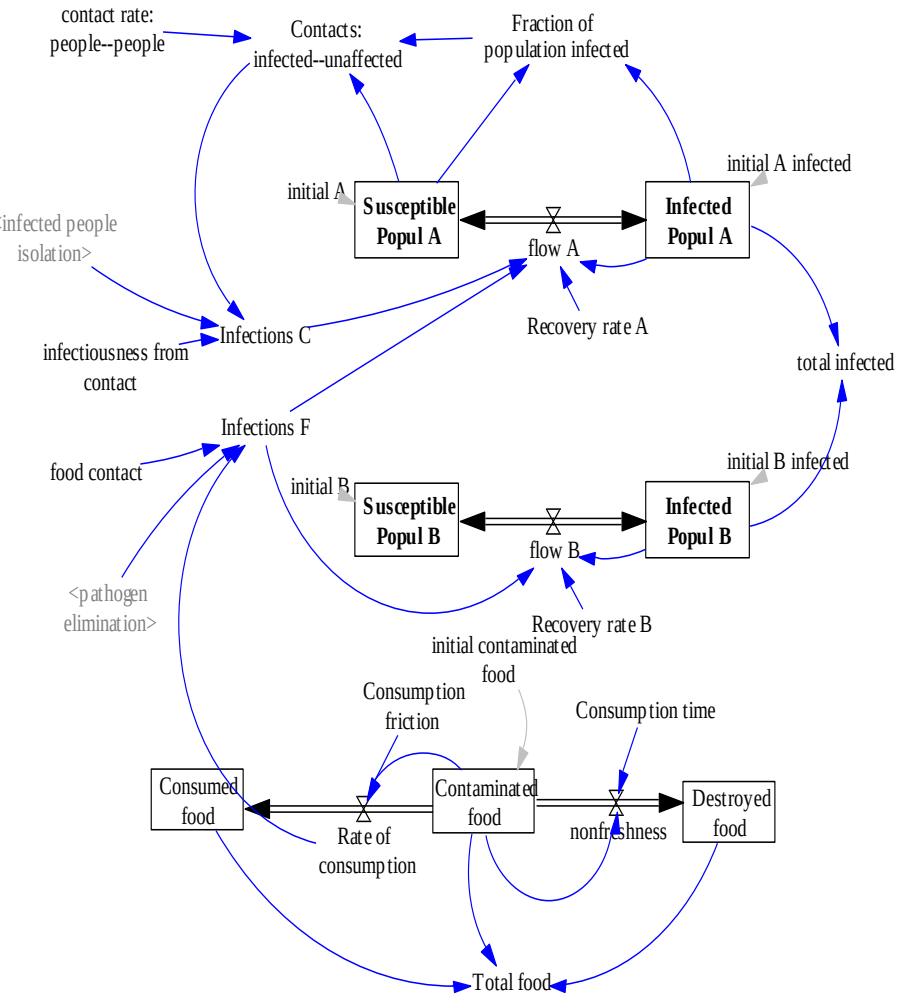

Fig. 2 The part of the model corresponding to the extended SIR model of an epidemic

```
rate of new infected people inspections =
  people inspection coefficient*( IF THEN ELSE( INTEGER
    (MAX (new loans for infected people inspections/cost of people
      inspection,0)) > new planned infected people inspections,
    new planned infected people inspections,
    INTEGER (MAX (new loans for infected people inspections/cost of people
      inspection,0) ) ))
```

The formula expresses the rule of undertaking new inspections on the basis of the necessary inspections (new planned infected people inspections) and available funds (Available funds) and the cost of inspections (cost of inspection).

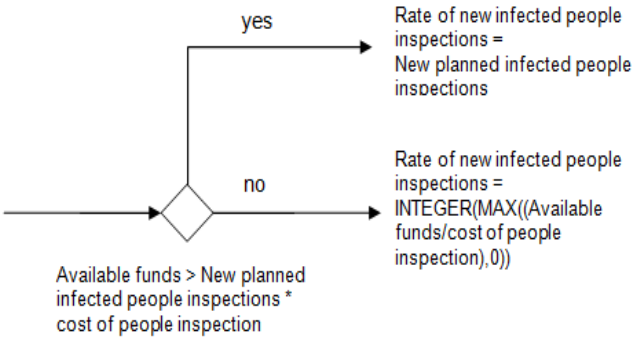

Fig. 1 The formula for the variable rate of new infected people inspections

The discussed model has 67 variables from which 21 are differential. The simulation of the model is in fact the numerical integration of differential-algebraic equations of the model. In order to avoid integrating differential-algebraic equations the model equations are transformed into differential equations (that the transformation is possible is remarkable feature of the Forrester's methodology), then explicit Euler, or Runge-Kutta schemes can be used to integrate the resulting ordinary differential equations. The result of model simulation is shown in Fig.4.

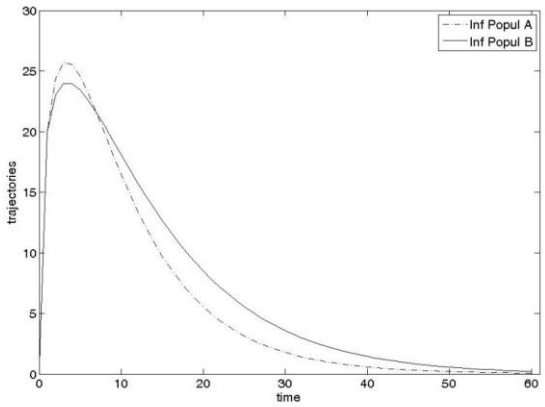

Fig. 4 Simulation results of the Forrester's model with decision rules

3. OPTIMIZATION COMPONENT OF THE SYSTEM

The optimization component serves two purposes. Firstly, it enables the calibration of the model. Secondly, it can be used to optimize the sanitary teams activities.

3.1. Optimization of sanitary teams activities

Our system can be used to optimize activities of sanitary teams. This is achieved by building the optimization problem on the basis of the model of an epidemic and sanitary teams activities. The algebraic equations are eliminated from the model and the model is described by only differential equations – that is done by DOML File Generator component of our system. The component writes also these equations according to the DOML format. DOML format is recognized by the DOML (Dynamic Optimization Modelling Language) compiler which is employed by the IDOS server where the script of the problem is sent in order to find a solution to the problem. The DOML language is based on the extension of the Modelica standard created for simulation purposes (Modelon, 2012). DOML File Generator gives only the partial description of the optimization problem: the definition of parameters of the problem; differential equations of the problem. The other part of the problem description: control variables (decision variables); objective function; constraints, can be added with the help of Optimizer GUI component of the system.

The DOML script of the optimization problem is sent to IDOS server (Pytlak, Tarnawski, Fajdek, Stachura, 2013) in order to accomplish the solution of the problem. We decided to use some kind of outsourcing since IDOS server is a dedicated server for dynamic optimization which is equipped with many solvers, furthermore it uses automatic differentiation tools so derivatives of optimization models do not have to be provided.

3.2. Example – dynamic optimization

In order to illustrate how IDOS Optimizer can be used we constructed the optimization problem based on the simulation model presented in the previous section. We eliminated decision rules from the simulation model (by modifying right-hand sides of some differential equations – as the result only part of the simulations model has changed – see Fig. 6. Furthermore, we introduced to the model control variables indicated as 'input' variables, which also presents essential parts of the DOML script related to the problem. In order to minimize the number of infected people (from both population A and B) new differential variables were inserted into the model (together with the corresponding differential equations – the last two equations in 'equation' section of the optimization model from Fig. 8):

```
der(Total_Inf_Pop_A) - Inf_Pop_A = 0
der(Total_Inf_Pop_B) - Inf_Pop_B = 0.
```

The optimization model has several constraints (inequality constraints which must be satisfied at every time) which are stated in 'constraint' section of the model. The first two equations are imposed in order to have properly defined measures for the effectiveness of sanitary teams activities. Among the constraints we also

have the constraint which says that the cost of sanitary teams activities cannot exceed some pre-specified amount. The presented optimization model had 21 ordinary differential equations, the optimization horizon was set to [0,60].

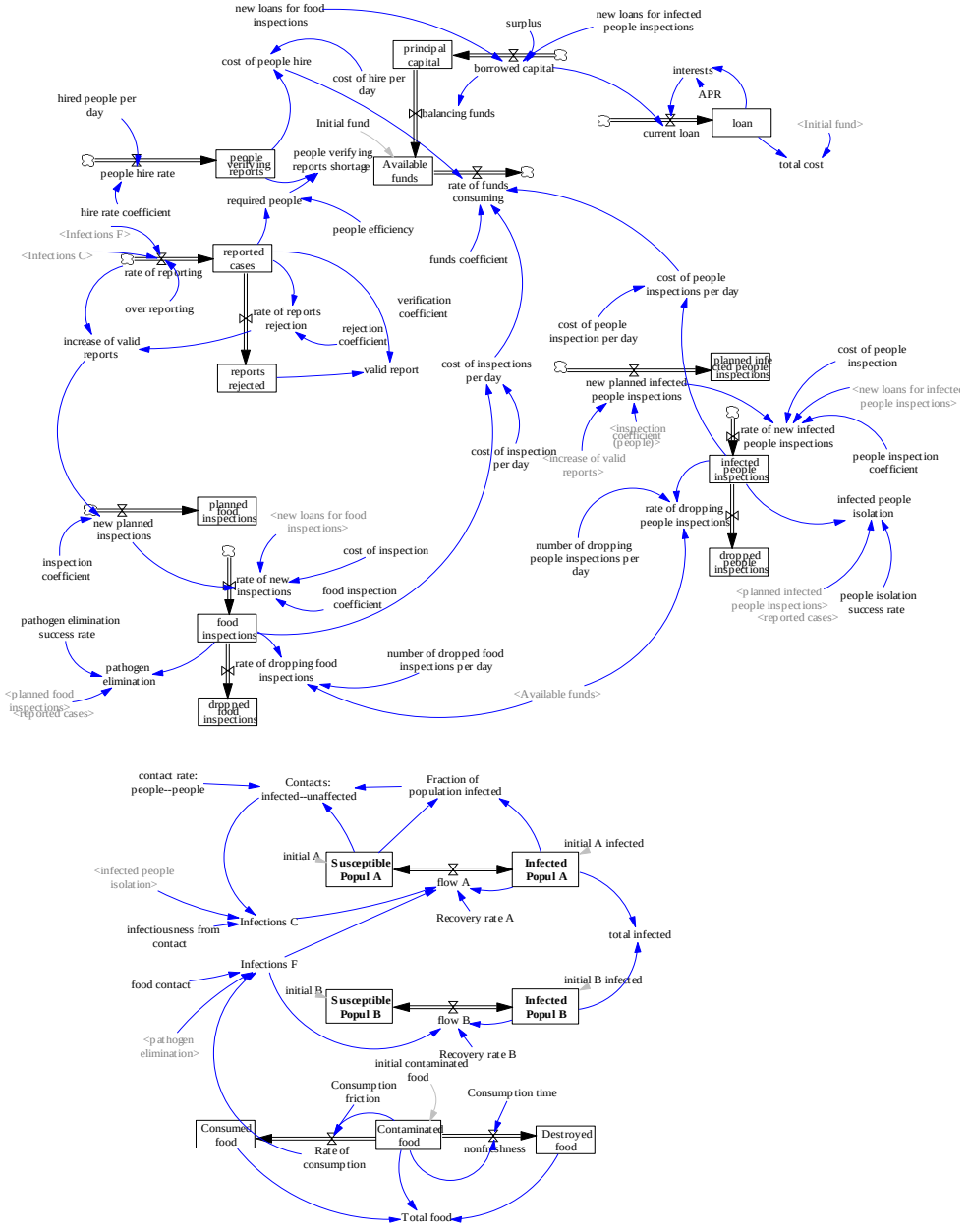

Fig. 5 The Forrester's model with control variables

```
package Epidemic_olado
optimization Epidemic_olado_opt(
objective = Total_Inf_Pop_A(finalTime) + Total_Inf_Pop_B(finalTime),
startTime = 0.0,
finalTime = 60.0)

parameter Real cost_of_food_inspections = 100;
parameter Real surplus = 500;
parameter Real cost_of_infected_people_inspection = 100;
parameter Real funds_coefficient = 1;
parameter Real cost_of_hire_per_day = 0.03;
.                                          . .    .    .    .    .
parameter Real recovery_rate_A = 0.2;
parameter Real recovery_rate_B = 0.1;
parameter Real Consumption_time = 15;
Real principal_capital(start = 0.0);
Real available_funds(start = 50.0);
Real people_veryfing_reports(start = 1.0);
Real loan(start = 0.0);  Real reported_cases(start = 0.0);  Real reports_rejected(start = 0.0);
Real planned_infected_people_inspections(start = 0.0);
Real infected_people_inspections(start = 0.0);
Real dropped_people_inspections(start = 0.0);
Real planned_food_inspections(start = 0.0);
Real food_inspections(start = 0.0);
Real dropped_food_inspections(start = 0.0);
Real Susc_Pop_A(start = 1000.0);
Real Inf_Pop_A(start = 0.0);  Real Susc_Pop_B(start = 200.0);  Real Inf_Pop_B(start = 0.0);
Real Contaminated_food(start = 800.0);
Real Destroyed_food(start = 1.0);  Real Consumed_food(start = 0.0);
Real Total_Inf_Pop_A(start = 0.0); Real Total_Inf_Pop_B(start = 0.0);
input Real hired_people_per_day (initialGuess = 5, min=0, max=30);
input Real new_food_inspections_per_day (initialGuess = 5, min=0, max=30);
input Real new_infected_people_inspections_per_day (initialGuess= 5, min=0, max=30);
input Real number_of_dropped_food_inspections_per_day (initialGuess= 5, min=0, max=30);
input Real number_of_dropping_people_inspections_per_day (initialGuess= 5, min=0, max=30);

annotation(solver="olado", steps="60", solver_option="method=17",
solver_option="inifile=olado.ini",solver_option="eps=1e-6", solver_option="max_iter=1000");

equation
der(principal_capital) - (((cost_of_food_inspections) * (new_food_inspections_per_day)) + (surplus)) +(
(cost_of_infected_people_inspection) * (new_infected_people_inspections_per_day))) -
((((cost_of_food_inspections) * (new_food_inspections_per_day)) + (surplus)) +
((cost_of_infected_people_inspection) * (new_infected_people_inspections_per_day)))=0;

der(available_funds) - ((((cost_of_food_inspections) * (new_food_inspections_per_day)) + (surplus)) +
((cost_of_infected_people_inspection) * (new_infected_people_inspections_per_day))) -
((((funds_coefficient) * ((cost_of_hire_per_day) * (people_veryfing_reports))) + ((food_inspections) *
(cost_of_inspection_per_day))) + ((cost_of_people_inspection_per_day) * (infected_people_inspections)))=0;

der(people_veryfing_reports) - ((hire_rate_coefficient)*(hired_people_per_day))-(0)=0;
```

Fig. 6 The DOML file of the optimizations problem

```
der(loan) - (((((cost_of_food_inspections) * (new_food_inspections_per_day)) + (surplus)) +
((cost_of_infected_people_inspection) * (new_infected_people_inspections_per_day))) + ((APR)*(loan)))-
(((cost_of_food_inspections) * (number_of_dropped_food_inspections_per_day)) +
((cost_of_infected_people_inspection) * (number_of_dropping_people_inspections_per_day)))=0;

 der(reported_cases) - ((over_reporting) * ((((((contact_rate_people_people) * (Susc_Pop_A)) *
((Inf_Pop_A) / ((Inf_Pop_A) + (Susc_Pop_A)))) * (infectiousness_from_contact)) * ((1)-
(((people_isolation_success_rate) * (infected_people_inspections)) /
((planned_infected_people_inspections) + (1))))) + ((((Consumption_friction) * (Contaminated_food)) *
(food_contact))*((1)-(((pathogen_elimination_succes_rate) * (food_inspections)) /
((planned_food_inspections)+(1)))))))-((rejections_coefficient)*(reported_cases))=0;

 der(reports_rejected) - ((rejections_coefficient)*(reported_cases))-(0)=0;
 der(planned_infected_people_inspections) - ((((over_reporting) * ((((((contact_rate_people_people) *
(Susc_Pop_A)) * ((Inf_Pop_A)/((Inf_Pop_A) + (Susc_Pop_A)))) * (infectiousness_from_contact))*((1)-
(((people_isolation_success_rate) * (infected_people_inspections)) /
((planned_infected_people_inspections) + (1))))) + ((((Consumption_friction) * (Contaminated_food)) *
(food_contact))*((1)-(((pathogen_elimination_succes_rate) * (food_inspections)) /
((planned_food_inspections) + (1)))))))-((rejections_coefficient) * (reported_cases))) *
(inspection_coefficient_people))-(0)=0;

der(infected_people_inspections) - ((people_inspection_coefficient) *
(new_infected_people_inspections_per_day))-(number_of_dropping_people_inspections_per_day)=0;
der(dropped_people_inspections) - (number_of_dropping_people_inspections_per_day)-(0)=0;

 der(planned_food_inspections) - ((inspections_coefficient) * (((over_reporting) *
((((((contact_rate_people_people) * (Susc_Pop_A)) * ((Inf_Pop_A) / ((Inf_Pop_A) + (Susc_Pop_A)))) *
(infectiousness_from_contact))*((1)-(((people_isolation_success_rate) * (infected_people_inspections)) /
((planned_infected_people_inspections) + (1))))) + ((((Consumption_friction) * (Contaminated_food)) *
(food_contact))*((1)-(((pathogen_elimination_succes_rate) * (food_inspections)) /
((planned_food_inspections)+(1)))))))-((rejections_coefficient)*(reported_cases))))-(0)=0;

der(Inf_Pop_A) - (((((((contact_rate_people_people) * (Susc_Pop_A)) * ((Inf_Pop_A)/((Inf_Pop_A) +
(Susc_Pop_A)))) * (infectiousness_from_contact))*((1) - (((people_isolation_success_rate) *
(infected_people_inspections)) / ((planned_infected_people_inspections) + (1))))) +
((((Consumption_friction) * (Contaminated_food)) * (food_contact)) * ((1)-
(((pathogen_elimination_succes_rate) * (food_inspections)) / ((planned_food_inspections) + (1)))))) -
((recovery_rate_A)*(Inf_Pop_A)))-(0)=0;
.                                   . . . . . .
der(Inf_Pop_B) - (((((Consumption_friction) * (Contaminated_food)) * (food_contact))*((1) –
(((pathogen_elimination_succes_rate) * (food_inspections)) / ((planned_food_inspections)+(1))))) -
((recovery_rate_B)*(Inf_Pop_B)))-(0)=0;

 der(Contaminated_food) -(((Contaminated_food)/(Consumption_time)) +
((Consumption_friction)*(Contaminated_food)))=0;
 der(Destroyed_food) - ((Contaminated_food)/(Consumption_time))-(0)=0;
 der(Consumed_food) - ((Consumption_friction)*(Contaminated_food))-(0)=0;
 der(Total_Inf_Pop_A) - Inf_Pop_A =0;  der(Total_Inf_Pop_B) - Inf_Pop_B =0;

 constraint
   infected_people_inspections-planned_infected_people_inspections <=1;
   food_inspections- planned_food_inspections <=1;  -Inf_Pop_A <=0;   -Inf_Pop_B <=0;   loan - 50000.0
<=0;
```

Fig. 7 The DOML file of the optimizations problem

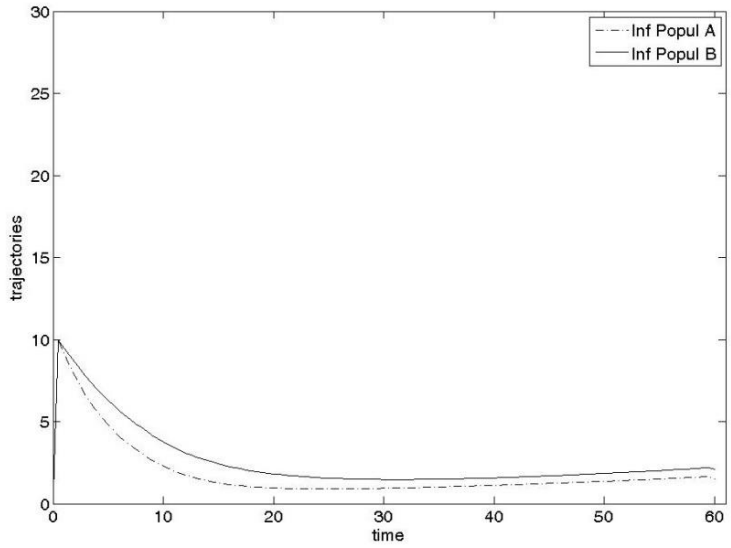

Fig. 8 Dynamic optimization results

The problem was solved by IPOPT optimization package ([13,14]) which is available at IDOS server within olado library. The result of optimization is shown in Fig.9. As one could expect to the number infected people in this case is much smaller than in the case of the simulation model which uses the decision rules. The fact that the number of infected people does not go to zero at the end of the horizon follows from the fact that we assumed in the epidemic model (see Fig. 3) that recovered people can become again susceptible people.

3.3. Example – CLP optimization

Our dynamic optimization problem related to sanitary inspection teams activities can also be solved by Constraint Logic Programming (CLP) (Fruwirth, Abdennadher,2003) methods that are used for solving complicated optimization tasks when the constraints consist not only of the algebraic equations. CLP methods are mainly applied to combinatorial problems, scheduling problems, problems from the areas of biotechlonogy or biology. The SIR / SEIR models are examples of that kind of the problems.

The most important feature of CLP models is their declarativity – the possibility of modeling the problem without showing the associated control flows. CLP techniques are the language based on the CSP (Constraint Satisfation Problems) methods. The constraints depend on their domains. The most common used domains are integer and real numbers.

```
optimize_SIR(SimTime) :-

  /* SIR state variables */
  dim(ConsumedFood,[NbPeriods]),
  dim(ContaminatedFood,[NbPeriods]),
  dim(DestroyedFood,[NbPeriods]),
  …
  /* initial values */
  ContaminatedFood[1] $= 9000.0,
  ConsumedFood[1]     $= 0.0,
  …
  /* constraints FOOD */
  ( for(T,1,NbPeriods), param(RateOfConsumption,ConsumptionFriction,ContaminatedFood) do
      RateOfConsumption[T] $= eval(ConsumptionFriction*ContaminatedFood[T])   ),
  ( for(T,1,NbPeriods), param(Nonfreshness,ConsumptionTime,ContaminatedFood) do
      Nonfreshness[T] $= eval(ContaminatedFood[T]/ConsumptionTime)   ),
  ( for(T,1,NbPeriods), param(TotalFood,ContaminatedFood,DestroyedFood,ConsumedFood) do
      TotalFood[T] $= ContaminatedFood[T] + DestroyedFood[T] + ConsumedFood[T]     ),
  ( for(T,1,NbPeriods-1), param(ConsumedFood,RateOfConsumption) do
      eval(ConsumedFood[T+1]) - eval(ConsumedFood[T]) $= eval(RateOfConsumption[T])   ),
  ( for(T,1,NbPeriods-1), param(DestroyedFood,Nonfreshness) do
      eval(DestroyedFood[T+1]) - eval(DestroyedFood[T]) $= eval(Nonfreshness[T])   ),
  ( for(T,1,NbPeriods-1), param(ContaminatedFood,Nonfreshness,RateOfConsumption) do
      eval(ContaminatedFood[T+1]) - eval(ContaminatedFood[T]) $=  - eval(Nonfreshness[T]) -
eval(RateOfConsumption[T])   ),

  /* constraints Infected B Group */
  ( for(T,1,NbPeriods), param(InfectionsF,RateOfConsumption,FoodContact,PathogenElimination) do
      InfectionsF[T] $= RateOfConsumption[T]*FoodContact*(1-PathogenElimination[T])   ),
  ( for(T,1,NbPeriods), param(FlowB,InfectionsF,RecoveryRateB,InfectedPopulB) do
      FlowB[T] $= InfectionsF[T] - RecoveryRateB * InfectedPopulB[T]   ),
  ( for(T,1,NbPeriods-1), param(SusceptiblePopulB,FlowB) do
      SusceptiblePopulB[T+1] - SusceptiblePopulB[T] $= -1 * eval(FlowB[T])   ),
  ( for(T,1,NbPeriods-1), param(InfectedPopulB,FlowB) do
      %InfectedPopulB[T+1] - InfectedPopulB[T] $=< eval(FlowB[T]),
      InfectedPopulB[T+1] - InfectedPopulB[T] $= eval(FlowB[T])   ),

  /* constraints Infected A Group */
  ( for(T,1,NbPeriods), param(FractionOfPopulationInfected,InfectedPopulA,SusceptiblePopulA) do
      FractionOfPopulationInfected[T] $= InfectedPopulA[T]/(InfectedPopulA[T]+SusceptiblePopulA[T])
),
  ( for(T,1,NbPeriods),
param(ContactsInfectedUnaffected,ContactRatePeoplePeople,SusceptiblePopulA,FractionOfPopulationInfec
ted) do
      ContactsInfectedUnaffected[T] $= ContactRatePeoplePeo-
ple*SusceptiblePopulA[T]*FractionOfPopulationInfected[T]   ),
  ( for(T,1,NbPeriods),
param(InfectionsC,ContactsInfectedUnaffected,InfectiousnessFromContact,InfectedPeopleIsolation) do
      InfectionsC[T] $= ContactsInfectedUnaffected[T]*InfectiousnessFromContact*(1-
InfectedPeopleIsolation[T])   ),
  ( for(T,1,NbPeriods), param(FlowA,InfectionsC,InfectionsF,RecoveryRateA,InfectedPopulA) do
      FlowA[T] $= (InfectionsC[T] + InfectionsF[T]) - (RecoveryRateA*InfectedPopulA[T])     ),
  ( for(T,1,NbPeriods-1), param(SusceptiblePopulA,FlowA) do
      SusceptiblePopulA[T+1] - SusceptiblePopulA[T] $= -1 * (FlowA[T])   ),
  ( for(T,1,NbPeriods-1), param(InfectedPopulA,FlowA) do
      InfectedPopulA[T+1] - InfectedPopulA[T] $= FlowA[T]   ),
```

Fig. 10 CLP file of the optimization problem

The basic searching methods of CLP are based on the constraint propagation and the most common CLP languages are derived from Prolog language (CHIP, Eclipse, SICStus Prolog). Some modeling aspects of CLP are described in (Wallace,2005), (Wallace,Schimpf,2002). In this paper we apply mechanisms of CLP of the ILOG constraint programming solver. ILOG uses the OPL language which is the native format.

Below we present an example of the script for solving a part of optimization where one minimizes only a number of infected people taking into account an infected food and omitting the sanitary team activities.

The presented problem minimize the total number of infected people of groups A and B. CLP optimization method is based upon branch and bound tree.

4. CONCLUSIONS

The paper presents the decision support systems for sanitary teams activities. There several novel features of the system. The system enables to model not only the spread of epidemic but also activities of sanitary teams. The system enables both model simulation and optimization of the related optimization model. The modeling component of the system employs the Forrester's methodology which was successfully used in the context of complex processes. The system employs cloud computing to perform optimization.

ACKNOWLEDGEMENTS

The model was constructed as a task „Dynamic modelling of the sanitary teams activities" for the project „Simulations and dynamic models" – grant number PBS1/A7/6/2012, project of Military University of Technology donated by NCBiR.

REFERENCES

Pytlak, R., Tarnawski, T., Fajdek, B., Stachura, M., (2013), Interactive Dynamic Optimization Server (IDOS) – Connecting one Modeling Language with Many Solvers, Optimization Methods & Solvers, in print.

Anderson, R.M., May, editors, (1991), Infectious Diseases of Humans: Dynamics and Control, Oxford University Press, Oxford.

Bailey, N.T.J, (1975), "The Mathematical Theory of Infectious Diseases", Griffin, London.

Fruwirth T., Abdennadher S., (2003) Essentials of constraint programming, Springer.

Kermack, W.O. , McKendrick, A.G., (1927), Contributions to the mathematical theory of epidemics, Proc. R. Soc. Lond., A 1927, 115: 700–721.

Kermack, W.O. , McKendrick, A.G., (1932), Contributions to the mathematical theory of epidemics, Proc. R. Soc. Lond., A 1932, 138: 55–83.

Kermack, W.O. , McKendrick, A.G., (1933), Contributions to the mathematical theory of epidemics, Proc. R. Soc. Lond., A 1933, 141: 94–122.

Murray, J.D., (1993), Mathematical biology. I. An introduction, Springer-Verlag, Berlin, Heidelberg, New York.
Murray, J.D., (2001), Mathematical biology. II. Spatial models, Springer-Verlag, Berlin, Heidelberg, New York.
Pytlak, R., (1999), Numerical Methods for Optimal Control Problems with State Constraints, Lecture Notes in Mathematics 1707, Springer-Verlag.
Sterman, J.D., (2000), Business Dynamics, McGraw-Hill.
Modelon AB, Modelica, (2012), „JModelica User's Guide" v.1.8.
Wachter, A., (2002), An Interior Point Algorithm for Large-Scale Nonlinear Optimization with Applications in Process Engineering, PhD thesis, Carnegie Mellon University, Pittsburgh, PA, USA.
Wachter, A., (2010), An introduction to Ipopt: A tutorial for downloading, installing, and using Ipopt.
Wallace M., (2005), Hybrid algorithms, local search and ECLiPSe, CP Summer School
Wallace M., Schimpf J., (2002), Finding the right hybrid algorithm – A combinatorial meta-problem, Annals of Mathematics and Artificial Intelligence 34: 259–269.
Zawadzki, T., Pytlak, R., (2011), Extending System Dynamics Approach to Higher Index DAE's, Proceedings of the System Dynamics Society 2011 Conference, July 24-26,

BIOGRAPHICAL NOTES

Radosław Pytlak is a Professor at Institute of Automatic Control and Robotics of Warsaw University of Technology. He teaches: modeling, simulation and optimization of dynamical systems; numerical methods; theory and methods of nonlinear programming; scheduling. His interests lie mainly in dynamic optimization and in algorithms for large scale optimization problems. He is the author of two monographs published in Springer-Verlag: Numerical methods for optimal control problems with state constraints; Conjugate gradient algorithms in nonconvex optimization. He published several papers in leading journals on optimization such as: SIAM J. on Optimization; SIAM J. on Control and Optimization; Journal of Optimization Theory and Applications; Numerische Mathematik.

Wojciech Stecz is an assistant professor at Faculty of Cybernetics Military University of Technology. He teaches: modeling and simulation of systems; numerical methods; theory and methods of nonlinear programming and scheduling.