Modelling Complex Systems by Integration of Agent-Based and Dynamical Systems Models

Tibor Bosse, Alexei Sharpanskykh, and Jan Treur
Vrije Universiteit Amsterdam
Department of Artificial Intelligence
{tbosse, sharp, treur}@cs.vu.nl

1 Introduction

Existing models for complex systems are often based on quantitative, numerical methods such as Dynamical Systems Theory (DST) [Port and Gelder 1995]. Such approaches often use numerical variables to describe global aspects and specify how they affect each other over time. An advantage of such approaches is that numerical approximation methods and software are available for simulation.

Agent-based modelling approaches take into account the local perspective of a possibly large number of agents and their behaviours. They are usually based on qualitative, logical languages. An advantage of such approaches is that they allow logical analysis of relationships between different parts of a model, for example relationships between global and local properties of a (multi-agent) system. Moreover, declarative models can be specified using logic-based languages close to natural language. Such models can be analyzed at a high level of abstraction. Furthermore, automated support is available for manipulation and design of models.

Complex systems often involve both qualitative aspects and quantitative aspects that can be modelled by agent-based (logical) and DST-based approaches respectively. It is not easy to integrate both types of approaches in one modelling method. On the one hand, it is difficult to incorporate logical aspects in differential equations. On the other hand, logical, agent-based modelling languages, often are not able to handle real numbers and calculations. This paper shows an integrative approach to simulate and analyse complex systems, integrating quantitative, numerical and qualitative, logical aspects within one temporal specification language. In Section 2, this language (called LEADSTO) is described in detail, and is illustrated for a system of differential equations (a Predator-Prey model) applying methods from numerical analysis. Section 3 shows how quantitative and qualitative aspects can be combined within the same model. Section 4 demonstrates how relationships can be established between dynamics of basic mechanisms (described in LEADSTO) and global dynamics of a process (described in a super-language of LEADSTO). Finally, Section 5 is a discussion. An extended version of this paper with more details appeared as [Bosse, Sharpanskykh and Treur, 2008].

2 Modelling dynamics in LEADSTO

Dynamics can be modelled in different forms. Based on the area within Mathematics called calculus, the Dynamical Systems Theory [Port and Gelder 1995] advocates to model dynamics by continuous state variables and changes of their values over time, which is also assumed continuous. In particular, systems of differential or difference equations are used. However, not for all applications dynamics can be modelled in a quantitative manner as required for DST. Sometimes qualitative changes form an essential aspect of the dynamics of a process. For example, to model the dynamics of reasoning processes usually a quantitative approach will not work. In such processes states are characterised by qualitative state properties, and changes by transitions between such states. For such applications often qualitative, discrete modelling approaches are advocated, such as variants of modal temporal logic, e.g. [Meyer and Treur 2002]. However, using such methods, the more precise timing relations are lost.

For the LEADSTO language described in this paper, the choice has been made to consider the timeline as continuous, described by real values, but for state properties both quantitative and qualitative variants can be used. The approach subsumes approaches based on simulation of differential or difference equations, and discrete qualitative modelling approaches. In addition, the approach makes it possible to combines both types of modelling within one model. Moreover, the relationships between states over time are described by either logical or mathematical means, or a combination thereof. This will be explained in more detail in Section 2.1. As an illustration, in Section 2.2 it will be shown how a system of ordinary differential equations representing the classical Predator-Prey model can be modelled and simulated in LEADSTO.

2.1 The LEADSTO language

Dynamics is considered as evolution of states over time. The notion of state as used here is characterised on the basis of an ontology defining a set of properties that do or do not hold at a certain point in time. For a given (order-sorted predicate logic) ontology Ont, the propositional language signature consisting of all *state ground atoms* (or *atomic state properties*) based on Ont is denoted by APROP(Ont). The *state properties* based on a certain ontology Ont are formalised by the propositions that can be made (using conjunction, negation, disjunction, implication) from the ground atoms. A *state* S is an indication of which atomic state properties are true and which are false, i.e., a mapping S: APROP(Ont) → {true, false}.

To specify simulation models a temporal language has been developed. This language (the LEADSTO language [Bosse et al. 2007]) enables to model direct temporal dependencies between two state properties in successive states, also called *dynamic properties*. A specification of dynamic properties in LEADSTO format has as advantages that it is executable and that it can often easily be depicted graphically. The format is defined as follows. Let α and β be state properties of the form 'conjunction of atoms or negations of atoms', and e, f, g, h non-negative real numbers. In the LEADSTO language the notation $\alpha \twoheadrightarrow_{e,\,f,\,g,\,h} \beta$ (also see Figure 1), means:

If state property α holds for a certain time interval with duration g, then after some delay (between e and f) state property β will hold for a certain time interval of length h.

An example dynamic property in LEADSTO format is the following: "observes(agent_A, food_present) $\twoheadrightarrow_{2,\,3,\,1,\,1.5}$ beliefs(agent_A, food_present)". Informally, this example expresses the fact that, if agent A observes that food is present during 1 time unit, then after a delay between 2 and 3 time units, agent A will belief that food is present during 1.5 time units. In addition, the LEADSTO language allows using sorts, sorted variables, real numbers, and mathematical operations, such as in "has_value(x, v) $\twoheadrightarrow_{e,\ f,\ g,\ h}$ has_value(x, v*0.25)". A *trace* or *trajectory* γ over a state ontology Ont is a time-indexed sequence of states over Ont (with the real numbers as time frame). To specify that a certain event (i.e., a state property) holds at every state within a certain time interval, the predicate holds_during_interval(event, t1, t2) is used. Here event is some state property, t1 is the start of the interval and t2 is the end of the interval.

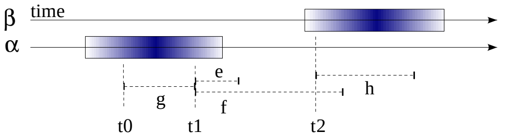


Figure. 1. Timing relationships in LEADSTO

2.2 Differential Equations in LEADSTO

Often behavioural models in the Dynamical Systems Theory are specified by systems of differential equations with given initial conditions for continuous variables and functions. One of the approaches to find solutions for such a system with given initial values is based on discretization, i.e., replacing a continuous model by a discrete one, whose solution is known to approximate that of the continuous one. For this methods of numerical analysis are usually used [Pearson 1986].

The simplest approach to approximate of solutions for ordinary differential equations is provided by Euler's method. To solve a differential equation of the form $dy/dt = f(y)$ with the initial condition $y(t_0)=y_0$ this method comprises a difference equation derived from the Taylor power series, until power 1: $y_{i+1}=y_i+h^* f(y_i)$, where $i\geq 0$ is the step number and $h>0$ is the integration step size.

This equation can be modelled in the LEADSTO language in the following way:

- States specify the respective values of y at different time points.
- The difference equation is modelled by a transition rule from current to successive state.
- The duration of an interval between state changes is defined by a step size h.

For the considered general case the LEADSTO model comprises the following rule:

$$\text{has\_value}(y, v1) \twoheadrightarrow_{0,\,0,\,h,\,h} \text{has\_value}(y, v1+h^* f(v1))$$

The initial value for the function y is specified by the following LEADSTO rule:

$$\text{holds\_during\_interval}(\text{has\_value}(y, y_0), 0, h)$$

By performing a simulation of the obtained model in the LEADSTO environment an approximate solution to the differential equation is found.

Although the first-order Euler method offers a stable solution, it is still rather rough and imprecise since the accumulated error within this method grows exponentially as the integration step size increases, therefore small step sizes are needed. To obtain more precise solutions for a given step size, higher order numerical methods are used. To illustrate higher-order numerical approaches, the fourth-order Runge-Kutta method is considered. This method is derived from a Taylor series up to the fourth order. It is known to be very accurate (the accumulated error is $O(h^4)$) and stable for a wide range of problems. The Runge-Kutta method for solving a differential equation of the form $dx/dt = f(t, x)$ is described by the following formulae:

$$x_{i+1} = x_i + h/6 *(k_1 + 2*k_2 + 2*k_3 + k_4),$$

where i≥0 is the step number, h>0 is the integration step size, and

$$k_1 = f(t_i, x_i) \quad k_2 = f(t_i + h/2, x_i + h/2 *k_1) \quad k_3 = f(t_i + h/2, x_i + h/2 *k_2) \quad k_4 = f(t_i + h, x_i + h* k_3).$$

To illustrate the proposed approach for simulations based on numerical methods, the system of ordinary differential equations representing the classical Lotka-Volterra model (a Predator-Prey model) [Morin 1999] is used. This model describes interactions between two species in an ecosystem, a predator and a prey. If $x(t)$ and $y(t)$ represent the number of preys and predators respectively, that are alive in the system at time t, then the Lotka-Volterra model is defined by:

$$dx/dt = a*x - b*x*y \quad dy/dt = c*b*x*y - e*y$$

where a is the per capita birth rate of the prey; b is a per capita attack rate; c is the conversion efficiency of consumed prey into new predators; e is the rate at which predators die in the absence of prey.

Now, using the Runge-Kutta method, the classical Lotka-Volterra model is described in the LEADSTO format as follows:

$$\text{has\_value}(x, v1) \wedge \text{has\_value}(y, v2) \twoheadrightarrow_{0, 0, h, h} \text{has\_value}(x, v1 + h/6 *(k_{11} + 2*k_{12} + 2*k_{13} + k_{14}))$$
$$\text{has\_value}(x, v1) \wedge \text{has\_value}(y, v2) \twoheadrightarrow_{0, 0, h, h} \text{has\_value}(y, v2 + h/6 *(k_{21} + 2*k_{22} + 2*k_{23} + k_{24})),$$

where $k_{11} = a*v1-b*v1*v2$, $k_{21} = c*b*v1*v2 - e*v2$, $k_{12} = a*(v1 + h/2 *k_{11}) - b*(v1 + h/2 *k_{11})*(v2 + h/2 *k_{21})$, $k_{22} = c*b*(v1 + h/2 *k_{11})*(v2 + h/2 *k_{21}) - e*(v2 + h/2 *k_{21})$, $k_{13} = a*(v1 + h/2 *k_{12}) - b*(v1 + h/2 *k_{12})*(v2 + h/2 *k_{22})$, $k_{23} = c*b*(v1 + h/2 *k_{12})*(v2 + h/2 *k_{22}) - e*(v2 + h/2 *k_{22})$, $k_{14} = a*(v1 + h *k_{13}) - b*(v1 + h *k_{13})*(v2 + h *k_{23})$, $k_{24} = c*b*(v1 + h *k_{13})*(v2 + h *k_{23}) - e*(v2 + h *k_{23})$.

The result of simulation of this model with the initial values $x_0=25$ and $y_0=8$ and the step size $h=0.1$ is given in [Bosse, Sharpanskykh and Treur, 2008]. It is identical to the result produced by Euler’s method with a much smaller step size ($h=0.01$) for the same example.

Although for most cases the Runge-Kutta method with a small step size provides accurate approximations, this method can still be computationally expensive and, in some cases, inaccurate. To achieve a higher accuracy together with minimum computational efforts, methods that allow the dynamic (adaptive) regulation of an integration step size are used. Generally, these approaches are based on the fact that the algorithm signals information about its own truncation error. The most commonly used technique for this is *step doubling* and *step halving*, see, e.g. [Gear 1971]. Since its format allows the modeller to include qualitative aspects, it is not difficult to incorporate step doubling and step halving into LEADSTO. See [Bosse, Sharpanskykh and Treur, 2008] for an illustration of how this can be done.

3 The Predator-Prey Model with Qualitative Aspects

In this section, an extension of the standard predator-prey model is considered, by some qualitative aspects of behaviour. Assume that the population size of both predators and preys within a certain eco-system is externally monitored and controlled by humans. Furthermore, both prey and predator species in this eco-system are also consumed by humans. A control policy comprises a number of intervention rules that ensure the viability of both species. Among such rules could be following:

- in order to keep a prey species from extinction, a number of predators should be controlled to stay within a certain range (defined by pred_min and pred_max);
- if a number of a prey species falls below a fixed minimum (prey_min), a number of predators should be also enforced to the prescribed minimum (pred_min);
- if the size of the prey population is greater than a certain prescribed bound (prey_max), then the size of the prey species can be reduced by a certain number prey_quota (cf. a quota for a fish catch).

These qualitative rules can be encoded into the LEADSTO simulation model for the standard predator-prey case by adding new dynamic properties and changing the existing ones in the following way:

$\text{has\_value}(x, v1) \wedge \text{has\_value}(y, v2) \wedge v1 < \text{prey\_max} \rightarrow_{0, 0, h, h} \text{has\_value}(x, v1+h*(a*v1-b*v1*v2))$

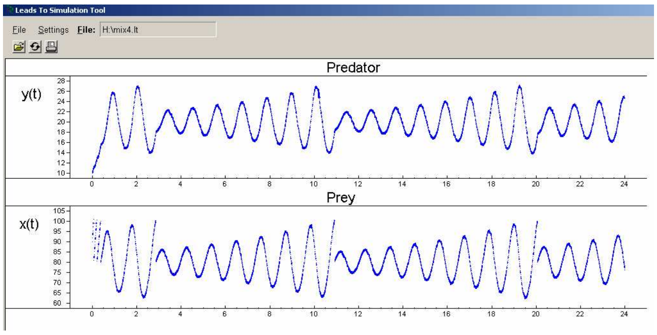
$\text{has\_value}(x, v1) \wedge \text{has\_value}(y, v2) \wedge v1 \geq \text{prey\_max} \rightarrow_{0, 0, h, h}$
$\quad \text{has\_value}(x, v1+h*(a*v1-b*v1*v2) - \text{prey\_quota})$

$\text{has\_value}(x, v1) \wedge \text{has\_value}(y, v2) \wedge v1 \geq \text{prey\_min} \wedge v2 < \text{pred\_max} \rightarrow_{0, 0, h, h}$
$\quad \text{has\_value}(y, v2+h* (c*b*v1*v2-e*v2))$

$\text{has\_value}(x, v1) \wedge \text{has\_value}(y, v2) \wedge v2 \geq \text{pred\_max} \rightarrow_{0, 0, h, h} \text{has\_value}(y, \text{pred\_min})$

$\text{has\_value}(x, v1) \wedge \text{has\_value}(y, v2) \wedge v1 < \text{prey\_min} \rightarrow_{0, 0, h, h} \text{has\_value}(y, \text{pred\_min})$

The result of simulation of this model using Euler's method with the parameter settings: a=4; b=0.2, c=0.1, e=8, pred_min=10, pred_max=30, prey_min=40, prey_max=100, prey_quota=20, x_0=90, y_0=10 is given in Figure 2.



File: H:\mix4.lt

Figure. 2. Simulation results for the Lotka-Volterra model combined with qualitative aspects.

4 Analysis In Terms of Local-Global Relations

Within the area of agent-based modelling, one of the means to address complexity is by modelling processes at different levels, from the global level of the process as a whole, to the local level of basic elements and their mechanisms. At each of these levels dynamic properties can be specified, and by interlevel relations they can be logically related to each other; e.g., [Sharpanskykh and Treur 2006]. These relationships can provide an explanation of properties of a process as a whole in terms of properties of its local elements and mechanisms. Such analyses can be done by hand or automatically. To specify the dynamic properties at different levels and their relations, a more expressive language is needed than simulation languages based on causal relationships, such as LEADSTO. To this end, the formal language TTL has been introduced as a super-language of LEADSTO; cf. [Bosse et al. 2006]. It is based on order-sorted predicate logic, and allows including numbers and arithmetical functions. Therefore most methods used in Calculus are expressible in this language, including methods based on derivatives and differential equations. In this section it is shown how to incorporate differential equations in the predicate-logical language TTL that is used for analysis. Further, in this section a number of global and local dynamic properties are identified, and it is shown how they can be expressed in TTL and logically related to each other.

Differential Equations in TTL

A differential equation of the form $dy/dt = f(y)$ with the initial condition $y(t_0)=y_0$ can be expressed in TTL on the basis of a discrete time frame (e.g., the natural numbers) in a straightforward manner:

$$\forall t\ \forall v\quad \text{state}(\gamma, t) \models \text{has\_value}(y, v) \quad \Rightarrow \quad \text{state}(\gamma, t{+}1) \models \text{has\_value}(y, v + h \bullet f(v))$$

The traces γ satisfying the above dynamic property are the solutions of the difference equation. However, it is also possible to use the dense time frame of the real numbers, and to express the differential equation directly. Thus, $x = dy/dt$ can be expressed as:

$$\forall t,w\ \ \forall \varepsilon{>}0\ \exists \delta{>}0\ \forall t',v,v'\ \ 0 < \text{dist}(t',t) < \delta\ \ \&\ \ \text{state}(\gamma, t) \models \text{has\_value}(x, w)\ \&$$
$$\text{state}(\gamma, t) \models \text{has\_value}(y, v)\ \&\ \ \text{state}(\gamma, t') \models \text{has\_value}(y, v') \Rightarrow \quad \text{dist}((v'-v)/(t'-t),w) < \varepsilon$$

where $\text{dist}(u,v)$ is defined as the absolute value of the difference. The traces γ for which this statement is true are (or include) solutions for the differential equation. Models consisting of combinations of difference or differential equations can be expressed in a similar manner. This shows how modelling constructs often used in DST can be expressed in TTL.

Global and Local Dynamic Properties

Within Dynamical Systems Theory, for global properties of a process more specific analysis methods are known. Examples of such analysis methods include mathematical methods to determine equilibrium points, the behaviour around equilibrium points, and the existence of limit cycles. Suppose a set of differential equations is given, for example a predator prey model: $dx/dt = f(x, y)$ and $dy/dt = g(x, y)$. Here, f(x, y) and g(x, y) are arithmetical expressions in x and y. Within TTL the following abbreviation is introduced as a definable predicate:

$$\text{point}(\gamma, t, x, v, y, w) \ \Leftrightarrow\ \text{state}(\gamma, t) \models \text{has\_value}(x, v) \wedge \text{has\_value}(y, w)$$

Equilibrium points

These are points in the (x, y) plane for which, when they are reached by a solution, the state stays at this point in the plane for all future time points. This can be expressed as a global dynamic property in TTL as follows:

$$\text{has\_equilibrium}(\gamma, x, v, y, w) \ \Leftrightarrow \forall t1\ [\ \text{point}(\gamma, t1, x, v, y, w) \ \Rightarrow\ \forall t2{\geq}t1\ \ \text{point}(\gamma, t2, x, v, y, w)\ ]$$
$$\text{occurring\_equilibrium}(\gamma, x, v, y, w) \ \Leftrightarrow \exists t\ \text{point}(\gamma, t, x, v, y, w)\ \&\ \ \text{has\_equilibrium}(\gamma, x, v, y, w)$$

Behaviour Around an Equilibrium

$\text{attracting}(\gamma, x, v, y, w, \varepsilon 0) \Leftrightarrow$

$\quad \text{has\_equilibrium}(\gamma, x, v, y, w)\ \&$

$\quad \varepsilon 0>0 \wedge \forall t\ [\ \text{point}(\gamma, t, x, v1, y, w1) \wedge \text{dist}(v1, w1, v, w) < \varepsilon 0 \Rightarrow$

$\quad \forall \varepsilon>0\ \exists t1 \geq t\ \forall t2 \geq t1\ [\ \text{point}(\gamma, t2, x, v2, y, w2) \Rightarrow \text{dist}(v2, w2, v, w) < \varepsilon\ ]\ ]$

Here, dist(v1, w1, v2, w2) denotes the distance between the points (v1, w1) and (v2, w2) in the (x, y) plane.

The global dynamic properties described above can also be addressed from a local perspective.

Local equilibrium property

From the local perspective of the underlying mechanism, equilibrium points are those points for which $dx/dt = dy/dt = 0$, i.e., in terms of f and g for this case $f(x, y) = g(x, y) = 0$.

$\text{equilibrium\_state}(v, w) \Leftrightarrow f(v, w) = 0\ \&\ g(v, w) = 0$

Local property for behaviour around an equilibrium:

$\text{attracting}(\gamma, x, v, y, w, \delta, \varepsilon 0, d) \Leftrightarrow \text{has\_equilibrium}(\gamma, x, v, y, w)\ \&$

$\varepsilon 0>0 \wedge 0< \delta <1 \wedge d \geq 0 \wedge \forall t\ [\ \text{point}(\gamma, t, x, v1, y, w1) \wedge \text{dist}(v1, w1, v, w) < \varepsilon 0 \Rightarrow$

$\forall t'\ [\ t+d \leq t' \leq t+2d\ \&\ \text{point}(\gamma, t', x, v2, y, w2) \Rightarrow \text{dist}(v2, w2, v, w) < \delta * \text{dist}(v1, w1, v, w)\ ]\ ]$

In terms of f and g, this can be expressed by relationships for the eigen values of the matrix of derivatives of f and g.

The properties of local and global level can be logically related by interlevel relations, for example, the following ones:

$\forall t\ [\ \text{state}(\gamma, t) \models \text{equilibrium\_state}(v, w) \Rightarrow \text{has\_equilibrium}(\gamma, x, v, y, w)$

$\exists d>0, \delta>0\ \text{attracting}(\gamma, x, v, y, w, \delta, \varepsilon 0, d) \Rightarrow \text{attracting}(\gamma, x, v, y, w, \varepsilon 0)$

5 Discussion

The LEADSTO approach proposed in this paper provides means to simulate models of dynamic systems that combine both quantitative and qualitative aspects. Sometimes such systems are called *hybrid systems* [Davoren and Nerode 2000]. In the control engineering area, hybrid systems are often considered as switching systems that represent continuous-time systems with isolated and often simplified discrete switching events [Liberzon and Morse 1999]. Yet in computer science the main interest in hybrid systems lies in investigating aspects of the discrete behaviour, while the continuous dynamics is often kept simple [Manna and Pnueli 1993].

Our LEADSTO approach provides as much place for modelling the continuous constituent of a system, as for modelling the discrete one. In contrast to many studies on hybrid systems in computer science (e.g., [Rajeev et al. 1997]), in which a state of a system is described by assignment of values to variables, in the proposed approach a state of a system is defined by (composite) objects using a rich ontological basis (i.e., typed constants, variables, functions and predicates). Furthermore, using TTL, a super-language of LEADSTO, analysis of dynamical systems by formalizing and applying standard techniques from the mathematical calculus can be performed.

Accuracy and efficiency of simulation results for hybrid systems provided by the proposed approach to a great extend depend on the choice of a numerical approximation method. Although the proposed approach does not prescribe usage of any specific approximation method (even the most powerful of them can be modelled in LEADSTO), for most of the cases the fourth-order Runge-Kutta method can be recommended, especially when the highest level of precision is not required. For simulating system models, for which high precision is demanded, higher-order numerical methods with an adaptive step size can be applied.

Bibliography

[1] Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A. & Treur, J., 2006. Specification and Verification of Dynamics in Cognitive Agent Models. In: Proceedings of the Sixth Int. Conf. on Intelligent Agent Technology, IAT'06. IEEE Computer Society Press, 247-255.

[2] Bosse, T., Jonker, C.M., Meij, L. van der, and Treur, J., 2007. A Language and Environment for Analysis of Dynamics by Simulation. International Journal of Artificial Intelligence Tools, 16, 435-464.

[3] Bosse, T., Sharpanskykh, A. & Treur, J., 2008. Integrating Agent Models and Dynamical Systems. In: Baldoni, M., Son, T.C., Riemsdijk, M.B. van, and Winikoff, M. (eds.). In Proc. of the 5th Int. Workshop on Declarative Agent Languages and Technologies, DALT'07, LNAI 4897, Springer Verlag, 50-68.

[4] Davoren, J.M. & Nerode, A., 2000. Logics for Hybrid Systems. In Proc. of the IEEE, 88, 7, 985-1010.

[5] Gear, C.W., 1971. Numerical Initial Value Problems in Ordinary Differential Equations. Englewood Cliffs, NJ: Prentice-Hall.

[6] Liberzon, D. & Morse, A. S., 1999. Basic problems in stability and design of switched systems, IEEE Control Systems Magazine, 19, 5, 59-70.

[7] Manna, Z. & Pnueli, A., 1993. Verifying Hybrid Systems. In Hybrid Systems, Lecture Notes in Computer Science 736, Springer-Verlag, 4-35.

[8] Meyer, J.J.Ch. & Treur, J. (eds.), 2002. Agent-based Defeasible Control in Dynamic Environments. Series in Defeasible Reasoning and Uncertainty Management Systems, 7. Kluwer Academic Publishers.

[9] Morin P.J, 1999. Community Ecology. Blackwell Publishing, USA.

[10] Pearson,C.E.,1986. Numerical Methods in Engineering and Science.CRC Press.

[11] Port, R.F., and Gelder, T. van (eds.), 1995. Mind as Motion: Explorations in the Dynamics of Cognition. MIT Press, Cambridge, Mass.

[12] Rajeev, A., Henzinger, T.A., & Wong-Toi, H., 1997. Symbolic analysis of hybrid systems. Proc. of 36th Annual Conf. on Dec. and Control, IEEE Press.

[13] Sharpanskykh, A. & Treur, J., 2006. Verifying Interlevel Relations within Multi-Agent Systems. Proc. of 17th European Conf. on AI, IOS Press, 290-294.