INFORMATICA, 1991, Vol.2, No.2, 255–277

APPLICATION OF LOGICAL PROGRAMMING FOR THE ANALYSIS OF AGGREGATIVE SPECIFICATIONS

Henrikas PRANEVITCHUS and Regina ČEPONYTĖ

Control Systems Department,
University of Technology,
233000, Kaunas, V.Juro St. 50, Lithuania.

Abstract. The paper considers the analysis technique of the general and individual properties of aggregative specifications. The method is based on constructing a set of axioms describing both the aggregate specifications and the properties of the model under investigation. The resolution method using logical programming language PROLOG is applied in creating the axiom system. An example of aggregative specification analysis for the alternating bit protocol is presented.

Key words: specification, aggregative approach, alternating bit protocol.

Introduction. Application of the aggregative approach and the method of control sequences for complex systems formalization and simulation is considered in (Pranevitchus, 1982). With the given approach the system under investigation is presented as a set of interacting piece-linear aggregates and the method of control sequences is used in the aggregative specification. The given method is used in creating systems of automation of aggregative simulation model building (Gorelik and Pranevitchus, 1985; Pranevitchus and Janilionis, 1985).

It is shown in (Pranevitchus and Chmieliauskas, 1983) that aggregative models can by used not only for building aggregative symulation models but for their correctness analysis as well. The method of reachable states is used for analysis. The given method is widely applied in analyzing computer network protocol correctness (Bochmann, 1987). In the given case the reachability graph is generated and later analyzed. Having completed the analysis one can determine such general properties as 1) deadlock freeness, 2) completeness, 3) termination or cyclic behavior, 4) boundednes, etc. The invariant approach has been created for the investigation of the individual properties of aggregative models (Pranevitchus and Panevėžys, 1988). An invariant is an assertion describing correct system functioning and remaining true in spite of the events taking place and of the transition from one state to another. The trueness of the invariant should be proved for every fragment related to the event. The correspondence between the aggregative model and the conceptual one can be checked by this method.

Special programs must usually by written for the analysis of the general and individual properties of aggregative models, for instance (Pranevitchus, Chmieliauskas and Pilkauskas, 1985). The given paper will present an aggregative model specification analysis approach based on constructing a set of axioms consisting of two parts. One of part of the axioms corresponds to the aggregative specifications, and the second part describes general and individual properties of the aggregative model under investigation. The consistency of the set of axioms created is checked by the resolution method using the logical programming language PROLOG.

1. Representing aggregative specifications by means of logical expressions. The given paragraph will deal with the predicates describing the aggregate state and the logical formulae, describing the operators of transitions and outputs resulting from internal and external events.

The piece-linear aggregate state at given moment is determined by continuous and discrete coordinates:

$$\text{state} = (w_1, \dots, w_f,\ d_1, \dots, d_n),$$

where $w_1, \dots, w_f$ – continuous coordinates, and $d_1, \dots, d_n$ – discrete coordinates.

The changes of the aggregate state result from occurrence of external and internal events. An output signal can by generated simultaneously with the event.

Therefore the following situations are possible:
1) an external event can by followed by a change of coordinates and an output signal can be generated;
2) an external event can by followed by a change of state without generating an output signal;
3) an internal event can by followed by a change of coordinates and an output signal is generated;
4) an internal event can by followed by a change of coordinates without generating an output signal.

Onset of an external event is linked with the arrival of an input signal at the corresponding input pole of the aggregate. Additional variables are introduced to describe the occurrence of external events: "input_p" – input pole and "input_v" – the value of the input signal.

Arrival of the input signal at the k-th pole of the aggregate Ag having the input signal value "sig" when the aggregate is in state "state_Ag= $(w_1, \dots, w_f,\ d_1, \dots, d_n)$" is described by a logical expression:

$$\begin{aligned}&\text{input\_p} = k \wedge \text{input\_v} = \text{sig}\\ &\qquad\qquad \wedge\ \text{state\_Ag} = (w_1, \dots, w_f,\ d_1, \dots, d_n).\end{aligned}$$

Let us determine the predicate QX_name(input_p, input_v, $w_1, \dots, w_f,\ d_1, \dots, d_f$) which indicates that with the signal arrival at the input pole the variables "input_p", "input_v" of the aggregate marked "name" are in the state "state=$(w_1, \dots, w_f,\ d_1, \dots, d_n)$".

Then the above given logical expression is described by the following predicate:

$$\text{QX\_Ag(k, sig, } w_1, \dots, w_f,\ d_1, \dots, d_f).$$

Similarly let the QY_name(output_p, output_v, $w_1, \dots, w_f,\ d_1, \dots, d_f$) represent the predicate describing the condition on which a signal of the value "output_v" is generated on the "output_p" pole of the aggregate marked "name".

The above situation will be represented by logical formula:

$$\text{QX\_name(input\_p, input\_v, } w_1, \dots, w_f,\ d_1, \dots, d_n) \wedge$$
$$\text{P}(w_1, \dots, w_f,\ d_1, \dots, d_n) \rightarrow$$
$$\text{QY\_name(output\_p, output\_v, next\_}w_1, \dots, \text{ next\_}w_f,$$
$$\text{next\_}d_1, \dots, \text{next\_}d_n),$$

here next_$w_1, \dots,$ next_w_f, next_$d_1, \dots,$ next_d_n are new values of the state coordinates and P$(w_1, \dots, w_f,\ d_1, \dots, d_n)$ the transition condition to be discused later.

The corresponding continuous coordinate having the value 1 is the condition for an internal event occurrence:

$$w_i = 1 \wedge \text{state} = (w_1, \dots, w_f,\ d_1, \dots, d_n).$$

Let this expression be represented by the predicate:

$$\text{QW\_name}(w_1, \dots, w_{i-1},\ 1,\ w_{i+1}, \dots, w_f,\ d_1, \dots, d_n).$$

Similarly the situations 2, 3, 4 will by represented by means of logical formula respectively:

$$\text{QX\_name(input\_p, input\_v, } w_1, \dots, w_f,\ d_1, \dots, d_n) \wedge$$
$$\text{P}(w_1, \dots, w_f,\ d_1, \dots, d_n) \rightarrow$$
$$\text{QW\_name(next\_}w_1, \dots, \text{next\_}w_f, \text{ next\_}d_1, \dots, \text{next\_}d_n),$$

$$\text{QW\_name}(w_1, \dots, w_{i-1},\ 1,\ w_{i+1}, \dots, w_f,\ d_1, \dots, d_n) \wedge$$
$$\text{P}(w_1, \dots, w_f,\ d_1, \dots, d_n) \rightarrow$$
$$\text{QY\_name}(\text{output\_}p,\ \text{output\_}v,\ \text{next\_}w_1, \dots, \text{next\_}w_f,$$
$$\text{next\_}d_1, \dots, \text{next\_}d_n),$$

$$\text{QW\_name}(w_1, \dots, w_{i-1},\ 1,\ w_{i+1}, \dots, w_f,\ d_1, \dots, d_n) \wedge$$
$$\text{P}(w_1, \dots, w_f;\ d_1, \dots, d_n) \rightarrow$$
$$\text{QW\_name}(\text{next\_}w_1, \dots, \text{next\_}w_f,\ \text{next\_}d_1, \dots, \text{next\_}d_n),$$

here input_$p = p,\ p \in P$, output_$p = o,\ o \in O,\ P$ and O a set of input and output poles of the aggregate marked "name".

In the aggregate system the aggregates are connected by communication channels. The communication of the channels and the aggregates system is described as the following form: 1) channel number; 2) an aggregate's name which generates an output signal on the "output_p" pole of the aggregate marked "name_1", 3) an aggregate's name on which arrives an input signal on the "input_p" pole of the aggregate marked "name_2".

The logical formula for every channel representing the interface of two aggregates marked "name_1" and "name_2" acquire the following form:

$$\text{QY\_name\_1}(\text{output\_}p, \text{output\_}v, w_1^1, \dots, w_{f_1}^1, d_1^1, \dots, d_{n_1}^1) \rightarrow$$
$$\text{QX\_name\_2}(\text{input\_}p,\ \text{input\_}v,\ w_1^2, \dots, w_{f_2}^2,\ d_1^2, \dots, d_{n_2}^2),$$

here $w_1^1, \dots, w_{f_1}^1,\ d_1^1, \dots, d_{n_1}^1;\ w_1^2, \dots, w_{f_2}^2,\ d_1^2, \dots, d_n^2$ are continuous and discrete coordinates of the aggregate states "name_1" and "name_2" respectively.

The change of the state coordinates in the aggregate specification is described by the transition and output operators.

Let us consider transition and output operators in detail. The description of these operators can comprise the following operations with the state coordinates:

1) dummy operator, skip, witch does not change the state coordinates. If the ratio Q describes the aggregate state, and $(w_1, \dots, w_f,\ d_1, \dots, d_n)$ is the state coordinates vector, the logical formula describing the transition is represented as:

$$Q(w_1, \dots, w_f,\ d_1, \dots, d_n) \to Q(w_1, \dots, w_f,\ d_1, \dots, d_n).$$

2) assignment operator:

$$\text{next\_}d_i := f(d_1, \dots, d_n), \quad i = \overline{1, n},$$

where $f(d_1, \dots, d_n)$ represents the change of the coordinate d_i. Next continuous coordinate $\text{next\_}w_i \in [0, 1]$, $j = \overline{1, f}$. The logical formula describing the assignment operator follows:

$$Q(w_1, \dots, w_f,\ d_1, \dots, d_n) \to$$
$$Q(\text{next\_}w_1, \dots, \text{next\_}w_f,\ \text{next\_}d_1, \dots, \text{next\_}d_n).$$

3) access operator has the following form:

if $R1$ then if $R2$ then $S1$,
else $S2$,
else if $R3$ then $S3$,
else $S4$,

here $S1, \dots, S4$ an assignment or a dummy operator $R1, \dots,$ $R3$ – logical expressions. The above given access statement can be transformed into the following form:

if $R1 \wedge R2$ then $S1$,
if $R1 \wedge \neg R2$ then $S2$,
if $\neg R1 \wedge R3$ then $S3$,
if $\neg R1 \wedge \neg R3$ then $S4$.

The access operator is represented by a logical formula as follows:

$$Q(w_1, \dots, w_f,\ d_1, \dots, d_n) \wedge R1 \wedge R2 \to$$
$$Q(\text{next\_}w_1, \dots, \text{next\_}w_f,\ \text{next\_}d_1, \dots, \text{next\_}d_n),$$

$$Q(w_1,\ldots,w_f,\ d_1,\ldots,d_n) \wedge R1 \wedge \neg R2 \rightarrow$$
$$Q(\text{next\_}w_1,\ldots,\text{next\_}w_f,\ \text{next\_}d_1,\ldots,\text{next\_}d_n),$$

$$Q(w_1,\ldots,w_f,\ d_1,\ldots,d_n) \wedge \neg R1 \wedge R3 \rightarrow$$
$$Q(\text{next\_}w_1,\ldots,\text{next\_}w_f,\ \text{next\_}d_1,\ldots,\text{next\_}d_n),$$

$$Q(w_1,\ldots,w_f,\ d_1,\ldots,d_n) \wedge \neg R1 \wedge \neg R3 \rightarrow$$
$$Q(\text{next\_}w_1,\ldots,\text{next\_}w_f,\ \text{next\_}d_1,\ldots,\text{next\_}d_n),$$

As the above given access operators are used in every transition and output operator the transition condition $P(w_1, \ldots, w_f,\ d_1,\ldots,d_n)$ represents logical expressions of the following type:

$$P(w_1,\ldots,w_f,\ d_1,\ldots,d_n) = \bigwedge_{i=1}^{n} R_i,$$

here n – is a number of logical expressions representing limitations of one or several coordinates.

2. A set of axioms for validation problem solution. In the previous chapter a form of logical formula, describing an aggregative specification is discussed. For validation problem solution additional formulae for the description of the properties under investigation must be introduced.

Investigation of general and individual properties is carried out at the global state of the set of aggregates. The global state includes the states of all the aggregates and is represented by the predicate:

$$\text{Q\_glob}(w_1^1,\ldots,w_{f_1}^1,\ d_1^1,\ldots,d_{n_1}^1,\ldots,$$
$$w_1^m,\ldots,w_{f_m}^m,\ d_1^m,\ldots,d_{n_m}^m),$$

here $n_i,\ f_i,\ i=\overline{1,m}$ the number of discrete and continuous coordinates of the i-th aggregate respectively.

Specification properties are investigated on a set of states limited by the initial and final states, of the form:

$$\begin{aligned}&\text{Q\_glob}(w_1^{10},\ldots,w_{f_1}^{10},\ d_1^{10},\ldots,d_{n_1}^{10},\ldots,\\&\qquad w_1^{m0},\ldots,w_{f_m}^{m0},\ d_1^{m0},\ldots,d_{n_m}^{m0}),\\&\text{Q\_glob}(w_1^{1t},\ldots,w_{f_1}^{1t},\ d_1^{1t},\ldots,d_{n_1}^{1t},\ldots,\\&\qquad w_1^{mt},\ldots,w_{f_m}^{mt},d_1^{mt},\ldots,d_{n_m}^{mt}),\end{aligned}$$

here the values of the initial and final states are denoted by indexes 0 and t respectively.

The general properties under investigation are:

1) Statistical dedlock freeness means that the system does not get into a state without output. In a dedlock state all the continuous coordinates equal zero. Dedlock state is determined by the predicate:

$$\text{Q\_glob}(0,\ldots,0,\ d_1^1,\ldots,d_{n_1}^1,\ldots,0,\ldots,0,\ d_1^m,\ldots,d_{n_m}^m),$$

2) Termination means that reaching a preset final state is guaranteed. The final state has been defined above.

3) Boundedness means that with the functioning of the system the values of the discrete coordinates do not exceed the present intervals. The limitations for every coordinate are represented by the following logical expressions:

$$\bigwedge_{j=1}^{n_i} d_j^i \in [a_j^i,\ b_j^i],\ i=\overline{1,m},$$

here a_j^i, b_j^i – are the limits of the variations of the j-th discrete coordinate of the i-th aggregate.

4) Completeness means that reception of all the possible messages is provided for in the specification. Non-feasibility of the completeness is represented by logical formula for every aggregate:

$$\text{QX\_name\_}i(\text{input\_}p,\ \text{input\_}v,\ w_1^i,\ldots,w_{f_i}^i,\ d_1^i,\ldots,d_{n_i}^i)\wedge$$
$$\text{input\_}v \notin X,$$

here $\notin$ indicates that the message accepted does not belong to the set X; X is the set of the possible input messages.
5) Absence of redundancy means that all the logical formulae of the specification are used in the specification analysis.

The invariant approach is used in the investigation of the individual properties of the system (Pranevitchus and Panevėžys, 1988). The invariant I represents the limitations of the global state coordinates and is of the form:

$$I = \bigvee_{i=1}^{l} P_i,$$

here P_i is the i-th disjunct representing limitations of one or several state coordinates, l is the number of disjuncts.

3. Example of representing aggregative specification by logical formulae. The alternating bit protocol specification including aggregative mathematical description and specification in the predicate logics language is presented below as an example of the use of logical representation of aggregative specification.

The structural scheme of aggregative model is presented in Fig. 1.

Aggregates and coordinates of aggregative model of the protocol are described below.

Aggregate A1 (sender). PSK (t_m) is the number of the reserved asknowledgements, Bit1 (t_m) is the value of the alternating bit in the last frame, that has already been sent out or formed, $w(e''_{11}, t_m)$ is the moment of the time at which the formation of the current frame is completed, $w(e''_{12}, t_m)$ is the moment of the time at which time-out over.

Aggregate A2 (transmission media). Bit2 (t_m) is the value of the alternating of the frame/acknowledgement being transmitted, $w(e''_{21}, t_m)$ is the moment of the time at which the acknowledgement will be transmitted (i.e., it will arrive at

the sender), $w(e''_{22}, t_m)$ is the moment of the time at which the frame will be transmitted (i.e., it will arrive at the receiver).
Aggregate A3 (receiver). Bit3 (t_m) is the value of the alternating bit in the last acknowledgement, KSK (t_m) is the number of the received frames, $w(e''_{31}, t_m)$ is the moment of time by which the acknowledgement will have been formed and sent out.

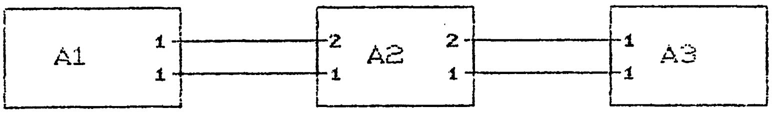

Fig. 1. Protocol aggregative scheme.

Description of transition and output operators are described below.

Aggregate A1:
1. A set of input signals : $X = \{x\}$, where x – input signal $x = (B)$, in which B is the value of the alternating bit in the acknowledgement.
2. A set of output signals : $Y = \{y\}$, where y – output signal $y = (B)$, in which B is the value of the alternating bit in the frame being transmitted.
3. A set of external events : $E' = \{e'_1\}$.
4. A set of internal events : $E'' = \{e''_{11},\ e''_{12}\}$.
5. Continuous component coordinate:
$W(t_m) = \{w(e''_{11},\ t_m),\ w(e''_{12},\ t_m)\}$.
6. Discrete component coordinate:
$\nu(t_m) = \{\mathrm{PSK}(t_m),\ \mathrm{Bit1}(t_m)\}$.
7. Initial state: $\mathrm{PSK}(t_m) = 0$, $\mathrm{Bit1}(t_m) = 1$, $w(e''_{11}, t_m) = 1$, $w(e''_{12}, t_m) = 0$.
8. Transition and output operators:
$H(e'_1)$:
if $(x = \mathrm{Bit1}(t_m))$ then $\mathrm{Bit1}(t_m) = \overline{\mathrm{Bit1}}(t_m)$;
$\mathrm{PSK}(t_m) = \mathrm{PSK}(t_m) + 1$;

$$w(e''_{11}, t_m) = 1;$$
$$w(e''_{12}, t_m) = 0;$$
else $w(e''_{11}, t_m) = 1;$
$$w(e''_{12}, t_m) = 0;$$

fi.

$H(e''_{11})$:

$$w(e''_{11}, t_m) = 1;$$
$$w(e''_{12}, t_m) = 0;$$

$G(e''_{11})$:

$$y = \text{Bit1}(t_m);$$

$H(e''_{12})$:

$$w(e''_{11}, t_m) = 1;$$
$$w(e''_{12}, t_m) = 0;$$

Aggregate A2:

1. A set of input signals : $X = \{x_1,\ x_2\}$, where x_1 – input signal $x_1 = (B)$, in which B is the value of the alternating bit in the frame being transmitted, x_2 input signal $x_2 = (B)$, in which B is the value of the alternating bit in the acknowledgement being transmitted.
2. A set of output signals : $Y = \{y_1,\ y_2\}$, where y_1 – output signal $y_1 = (B)$, in which B is the value of the alternating bit in the acknowledgement being transmitted, y_2 – output signal $y_2 = (B)$, in which B is the value of the alternating bit in the frame being transmitted,
3. A set of external events : $E' = \{e'_1,\ e'_2\}$.
4. A set of internal events : $E'' = \{e''_{21},\ e''_{22}\}$.
5. Continuous component coordinate:
$W(t_m) = \{w(e''_{21},\ t_m),\ w(e''_{22},\ t_m)\}$.
6. Discrete component coordinate : $\nu(t_m) = \{\text{Bit2}(t_m)\}$.
7. Initial state : $\text{Bit2}(t_m) = 0,\ w(e''_{21},\ t_m) = 0,$
$w(e''_{22},\ t_m) = 0$.
8. Transition and output operators:

$H(e'_1)$:

if $w(e''_{22},\ t_m) = 0;$ then $\text{Bit2}(t_m) = x_2;$

if $p_{21} > P$
then $w(e''_{21},\ t_m) = 0$;
else $w(e''_{21},\ t_m) = 1$;
fi.
$H(e'_2)$:
if $w(e''_{21},\ t_m) = 0$; then $\mathrm{Bit2}(t_m) = x_1$;
if $p_{22} > P$
then $w(e''_{22},\ t_m) = 0$;
else $w(e''_{22},\ t_m) = 1$;
fi.
$H(e''_{21})$:
$w(e''_{21},\ t_m) = 0$;
$G(e''_{21})$:
$y_1 = \mathrm{Bit2}(t_m)$;
$H(e''_{22})$:
$w(e''_{22},\ t_m) = 0$;
$G(e''_{22})$:
$y_2 = \mathrm{Bit2}(t_m)$;

Aggregate A3:

1. A set of input signals : $X = \{x\}$, where x – input signal $x = (B)$, in which B is the value of the alternating bit in the frame being received.
2. A set of output signals : $Y = \{y\}$, where y – output signal $y = (B)$, in which B is the value of the alternating bit in the acknowledgement being transmitted.
3. A set of external events : $E' = \{e'_1\}$.
4. A set of internal events : $E'' = \{e''_{31}\}$.
5. Continuous component coordinate : $W(t_m) = \{w(e''_{31},\ t_m)\}$.
6. Discrete component coordinate : $\nu(t_m) = \{\mathrm{KSK}(t_m), \mathrm{Bit3}(t_m)\}$.
7. Initial state : $\mathrm{KSK}(t_m) = 0$, $\mathrm{Bit3}(t_m) = 0$, $w(e''_{31},\ t_m) = 0$.
8. Transition and output operators:

$H(e'_1)$:

if $(x \neq \mathrm{Bit3}(t_m))$ then $\mathrm{Bit3}(t_m) = x$;
$\mathrm{KSK}(t_m) = \mathrm{KSK}(t_m) + 1$;
$w(e''_{31},\ t_m) = 1$;
else $w(e''_{31},\ t_m) = 1$;
$w(e''_{12},\ t_m) = 0$;
fi.
$H(e''_{31})$:

$$w(e''_{31}, t_m) = 0;$$

$G(e''_{31})$:

$$y = \mathrm{Bit3}(t_m);$$

Table. Table of aggregate interfacing

Channel	Aggregate name	Output	Aggregate name	Input
1.	A1	y	A2	x_2
2.	A2	y_2	A3	x
3.	A2	y_1	A1	x
4.	A3	y_3	A2	x_1

Transition and output operators for every aggregate will be represented by logical formulae. The notations follow:

$$\begin{aligned}
&w(e''_{12},\ t_m) = W12,\\
&w(e''_{21},\ t_m) = W21,\\
&w(e''_{22},\ t_m) = W22,\\
&w(e''_{31},\ t_m) = W31,\\
&\mathrm{Bit1}(t_m) = \mathrm{Bit1},\\
&\mathrm{Bit2}(t_m) = \mathrm{Bit2},\\
&\mathrm{Bit3}(t_m) = \mathrm{Bit3},\\
&\mathrm{PSK}(t_m) = \mathrm{PSK},\\
&\mathrm{KSK}(t_m) = \mathrm{KSK}.
\end{aligned}$$

Representation of the transition operator $H(e_1')$ by logical formulae will be shown in detail. The operators described in the transition can be represented as follows:
1. $\text{input\_}p = 1 \wedge \text{input\_}v = \text{Bit1} \rightarrow$
$\text{Bit1} = \overline{\text{Bit1}} \wedge \text{PSK} = \text{PSK} + 1 \wedge W11 = 1 \wedge W12 = 0.$
2. $\text{input\_}p = 1 \wedge \text{input\_}v \neq \text{Bit1} \rightarrow W11 = 1 \wedge W12 = 0.$
This means that at the signal arrival at the pole $\text{input\_}p = 1$ when the $\text{input\_}v$ =Bit1 the aggregate coordinates change according to the first description; if a signal of $\text{input\_}v$ $\neq$Bit1 arrives at the pole $\text{input\_}p = 1$, the coordinates change according to the second description.

The transition description, using the predicates introduced in Chapter 1, follow:

$$\text{QX\_A1}(1,\ \text{input\_}v,\ W11,\ W12,\ \text{Bit1}, \text{PSK}) \wedge \text{input\_}v = \text{Bit1} \rightarrow \text{QW\_A1}(1,\ 0,\ \text{Bit1},\ \text{PSK}+1),$$

$$\text{QX\_A1}(1,\ \text{input\_}v,\ W11,\ W12,\ \text{Bit1}, \text{PSK}) \wedge \text{input\_}v \neq \text{Bit1} \rightarrow \text{QW\_A1}(1,\ 0,\ \text{Bit1},\ \text{PSK}).$$

Similarly the other transition operators are described. The set of logical formulae of the whole aggregative specification is given below.

Aggregate A1.

$$\text{QX\_A1}(1,\ \text{input\_}v,\ \text{W11},\ \text{W12},\ \text{Bit1},\ \text{PSK}) \wedge \text{input\_}v = \text{Bit1} \rightarrow \text{QW\_A1}(1,\ 0,\ \text{Bit1},\ \text{PSK}+1), \tag{1}$$

$$\text{QX\_A1}(1,\ \text{input\_}v,\ \text{W11},\ \text{W12},\ \text{Bit1},\ \text{PSK}) \wedge \text{input\_}v \neq \text{Bit1} \rightarrow \text{QW\_A1}(1,\ 0,\ \text{Bit1},\ \text{PSK}), \tag{2}$$

$$\text{QW\_A1}(1,\ \text{W12},\ \text{Bit1},\ \text{PSK}) \rightarrow \text{QY\_A1}(1,\ \text{Bit1},\ 0,\ 1,\ \text{Bit1},\ \text{PSK}), \tag{3}$$

$$\text{QW\_A1(W11, 1, Bit1, PSK)} \rightarrow \text{QW\_A1(1, 0, Bit1, PSK)}. \tag{4}$$

Aggregate A2.

$$\text{QX\_A2(1, input\_}v\text{, W21, W22, Bit2)} \wedge \text{W22} = 0 \wedge p_{21} < P \rightarrow \text{QW\_A2(1, W22, Bit2)}, \tag{5}$$

$$\text{QX\_A2(1, input\_}v\text{, W21, W22, Bit2)} \wedge \text{W22} = 0 \wedge p_{21} > P \rightarrow \text{QW\_A2(0, W22, Bit2)}, \tag{6}$$

$$\text{QX\_A2(2, input\_}v\text{, W21, W22, Bit2)} \wedge \text{W21} = 0 \wedge p_{22} < P \rightarrow \text{QW\_A2(0, W21, 1, Bit2)}, \tag{7}$$

$$\text{QX\_A2(2, input\_}v\text{, W21, W22, Bit2)} \wedge \text{W21} = 0 \wedge p_{22} > P \rightarrow \text{QW\_A2(0, W21, 1, Bit2)}, \tag{8}$$

$$\text{QW\_A2(1, W22, Bit2)} \rightarrow \text{QY\_A2(1, Bit2, 0, W22, Bit2)}, \tag{9}$$

$$\text{QW\_A2(W21, 1, Bit2)} \rightarrow \text{QY\_A2(2, Bit2, W21, 0, Bit2)}. \tag{10}$$

Aggregate A3.

$$\text{QX\_A3(1, input\_}v\text{, W31, Bit3, KSK)} \wedge \text{input\_}v \neq \text{Bit3} \rightarrow \text{QW\_A3(1, input\_}v\text{, KSK)}, \tag{11}$$

$$\text{QW\_A3}(1,\ \text{Bit3},\ \text{KSK}) \to$$
$$\text{QY\_A3}(1,\ \text{Bit3},\ 0,\ \text{Bit3},\ \text{KSK}). \tag{12}$$

Table of aggregate interfacing:

$$\text{QY\_A1}(1,\ \text{output\_}v,\ \text{W11},\ \text{W12},\ \text{Bit1},\ \text{PSK}) \to$$
$$\text{QX\_A2}(2,\ \text{output\_}v,\ \text{W21},\ \text{W22},\ \text{Bit2}), \tag{13}$$

$$\text{QY\_A2}(1,\ \text{output\_}v,\ \text{W21},\ \text{W22},\ \text{Bit2}) \to$$
$$\text{QX\_A1}(1,\ \text{output\_}v,\ \text{W11},\ \text{W12},\ \text{Bit1},\ \text{PSK}), \tag{14}$$

$$\text{QY\_A2}(2,\ \text{output\_}v,\ \text{W21},\ \text{W22},\ \text{Bit2}) \to$$
$$\text{QX\_A3}(1,\ \text{output\_}v,\ \text{W31},\ \text{Bit3}), \tag{15}$$

$$\text{QY\_A3}(1,\ \text{output\_}v,\ \text{W31},\ \text{Bit3}) \to$$
$$\text{QX\_A2}(1,\ \text{output\_}v,\ \text{W21},\ \text{W22},\ \text{Bit2}). \tag{16}$$

Global initial state:

$$\text{Q\_glob}(1,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0). \tag{17}$$

Global final state:

$$\text{Q\_glob}(1,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 2,\ 2). \tag{18}$$

General properties:

Deadlock state:

$$\text{Q\_glob}(0,\ 0,\ 0,\ 0,\ 0,\ \text{Bit1},\ \text{Bit2},\ \text{Bit3},\ \text{PSK},\ \text{KSK}). \tag{19}$$

Unfeasible property boundedness:

$$\text{Q\_glob(W11, W12, W21, W22, W31,}\\ \text{Bit1, Bit2, Bit3, PSK, KSK)}\wedge\\ \text{Bit1} \notin [0,1] \wedge \text{Bit2} \notin [0,1] \wedge \text{Bit3} \notin [0,1]. \tag{20}$$

Unfeasible property of completeness:

$$\text{QX\_A1(input\_}p\text{, input\_}v\text{, W11, W12, Bit1, PSK)}\wedge\\ \text{input\_}v \notin [0,1] \tag{21}$$

$$\text{QX\_A2(input\_}p\text{, input\_}v\text{, W21, W22, Bit2)}\wedge\\ \text{input\_}v \notin [0,1] \tag{22}$$

$$\text{QX\_A(input\_}p\text{, input\_}v\text{, W31, Bit3, KSK)}\wedge\\ \text{input\_}v \notin [0,1] \tag{23}$$

Invariant unfeasibility:

$$\text{Q\_glob(W11, W12, W21, W22, W31,}\\ \text{Bit1, Bit2, Bit3, PSK, KSK)} \wedge \neg I. \tag{24}$$

The invariant I for a alternating bit is given in (Pranevitchus and Panevėžys, 1988).
If the logical formulae describing aggregative specifications and interfacing table are denoted A_p, initial state description $-A_I$, final state one $-A_0$, and the properties under investigation $-A_s$, then, having the set of axioms $A_p \wedge A_I \wedge A_0 \wedge A_s$, conclusions can be drawn about the problems formulated.

Considering the above alternating bit example we see that the set A_p includes logical formulae 1–16, A_I – represents 17, A_0 – 18, A_s – 19–24.

The following problem can be formulated : will a final state be reached from the initial one and simultaneously will the logical formulae determining general and individual properties be valid?.

4. Use of the language PROLOG for solving the validation problem. The problem formulated in the previous chapter can by solved by the method of resolution using the predicate logics language PROLOG based on logical programming. The language allows to use formal specifications in the predicate logics by means of PROLOG (Sidhu and Cral, 1988).

To epresent logical expressions in the language PROLOG the following parts of the program must be determined:

1. The part of descriptions:

    1.1. The part determining constants (CONSTANTS).

    1.2. The part determining date base (DATEBASE).

    1.3. The part determining the predicates (PREDICATES).

2. The part describing aggregates.

    2.1. The part describing transitions.

    2.1.1. Transitions of external events, e.g.:

$$\text{QX\_A1}(1,\ \text{Input\_}v,\ \text{W11},\ \text{W12},\ \text{Bit1},\ \text{PSK}) :-$$
$$\text{Input\_}v = \text{Bit1},\ \text{PSKn} = \text{PSK} + 1,\ \text{Bit1n} = \text{MOD}(\text{Bit1}),$$
$$\text{QW\_A1}(1,\ 0,\ \text{Bit1n},\ \text{PSKn}).$$

This means that at the signal arrival at the pole 1 when Input_v =Bit1, the state coordinates change, i.e., the discrete coordinates acquire new values PSK=PSK+1 and Bit1n= MOD (Bit1), the continuous coordinates W11= 1. W12= 0, MOD predicate determines the variation according to the modulus 2.

2.1.2. Transition of internal events, e.g.:

$$\text{QW\_A1}(1,\ \text{W12},\ \text{Bit1},\ \text{PSK}):-$$
$$\text{QY\_A1}(1,\ \text{Bit1},\ 0,\ 1,\ \text{Bit1},\ \text{PSK}).$$

This means that on occurrence of an internal event (condition W11=1) a signal of the value Bit1 is delivered to the output, and the continuous coordinates obtain the value W11= 0, W12= 1.

3. The part describing aggregate interfacing, e.g.:

$$\text{QY\_A1}(1,\ \text{Output\_}v,\ \text{W11},\ \text{W12},\ \text{Bit1},\ \text{PSK}):-$$
$$\text{Current\_state}(\text{W11},\ \text{W12},\ \text{W21},\ \text{W22},\ \text{W31},$$
$$\text{Bit1},\ \text{Bit2},\ \text{Bit3},\ \text{PSK},\ \text{KSK}),$$
$$\text{QX\_A2}(2,\ \text{Output\_}v,\ \text{W21},\ \text{W22},\ \text{Bit2}).$$

This means that the pole 1 of the aggregate A1 is connected by a channel with the pole 2 of the aggregate A2 and a signal of the value Output_v is transmitted. The predicate Current_state changes the values of the current state.

4. The part describing validation axioms.

4.1. Description of the general properties.

4.1.1. Dedlock state, e.g.:

$$\text{Q\_glob}(0,\ 0,\ 0,\ 0,\ 0,\ \_,\ \_,\ \_,\ \_,\ \_):-$$

write("deadlock state).

4.1.2. Boundedness property is not implemented, e.g.:

$$\text{Q\_glob}(\text{W11},\ \text{W12},\ \text{W21},\ \text{W22},\ \text{W31},$$
$$\text{Bit1},\ \text{Bit2},\ \text{Bit3},\ \text{PSK},\ \text{KSK}):-$$
$$\text{D}(\text{Bit1},\ \text{Bit2},\ \text{Bit3},\ \text{PSK},\ \text{KSK}),$$

write ("boundedness property is not implemented").

Here the predicate D acquires the TRUE if the boundedness property is not implemented.

4.1.3. Unfeasible completeness property, e.g.:

$$\mathrm{QX\_A1(Input\_}p,\ \mathrm{Input\_}v,\ \mathrm{W11,\ W12,\ Bit1,\ PSK) :-}$$
$$\mathrm{P(Input\_}v),$$

write("completeness property is not implemented").
Here is predicate P acquires the meaning TRUE if there is no logical formulae for the input signal Input_v.

4.2. Non-implementing of the invariant, e.g.:

$$\mathrm{Q\_glob(W11,\ W12,\ W21,\ W22,\ W31,}$$
$$\mathrm{Bit1,\ Bit2,\ Bit3,\ KSK) :-}$$
$$\mathrm{I(W11,\ W12,\ W21,\ W22,\ W31,\ Bit1,\ Bit2,\ Bit3,\ KSK),}$$

write ("the invariant is not implemented").
Here the predicate I acquires the meaning TRUE if the invariant is not implemented.

4.3. Initial state, e.g.:

$$\mathrm{Q\_glob(1,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0).}$$

4.4. Final state, e.g.:

$$\mathrm{Q\_glob(1,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 2,\ 2).}$$

5. Relation of the aggregate local state is the global one, e.g.:

$$\mathrm{QW\_A1(W11,\ W12,\ Bit1,\ PSK) :-}$$
$$\mathrm{Current\_state(W11,\ W12,\ W21,\ W22,\ W31,}$$
$$\mathrm{Bit1,\ Bit2,\ Bit3,\ PSK,\ KSK),}$$
$$\mathrm{Q\_glob(W11,\ W12,\ W21,\ W22,\ W31}$$
$$\mathrm{Bit1,\ Bit2,\ Bit3,\ PSK,\ KSK).}$$

```
→ 1+1-0-0-0-0-1-0-0
: :
: W11
: :
: → 2+0-1-0-1-0-1-1-0
: : :
: : W12
: : :
: : → 3+1-0-0-1-0-1-1-0
: : : :
: : : W11
: : : :
: : : : → 2+0-1-0-1-0-1-1-0
: : : : → 4+0-1-0-0-0-1-1-0
: : : : :
: : : : W12
: : : : :
: : : : → 5+1-0-0-0-0-1-1-0
: : : : : :
: : : : : W11
: : : : : :
: : : : : → 2+0-1-0-1-0-1-1-0
: : : : : → 4+0-1-0-0-0-1-1-0
: : → 3+1-0-0-1-0-1-1-0
: : : :
: : : W22
: : : :
: : : → 6+1-0-0-0-1-1-1-1
: : : : :
: : : : W11
: : : : :
: : : : → 7+0-1-0-1-1-1-1-1
: : : : : :
: : : : : W12
: : : : : :
: : : : : → 8+1-0-0-1-0-1-1-1
: : : : : :
: : : : : W11
: : : : : :
: : : : : → 7+0-1-0-1-1-1-1-1
: : : : : → 9+0-1-0-0-1-1-1-1
: : : : : : :
: : : : : : W12
: : : : : : :
: : : : : : → 6+1-0-0-0-1-1-1-1
: : : : : → 9+0-1-0-0-1-1-1-1
: : : : : : :
: : : : : : W31
: : : : : : :
: : : : : : → 10+0-1-1-0-0-1-1-1
```

Fig. 2. A fragment of the set of states.

6. Selection of an active aggregate, e.g.:

> Q_glob(W11, W12, W21, W22, W31,
> Bit1, Bit2, Bit3, PSK, KSK) : –
> IS(W11, W12, W21, W22, W31),
> QW_A1(W11, W12, Bit1, PSK).

Here the predicate IS selects the aggregate with a continuous coordinate equal to zero.

5. Validation results. The approach proposed was applied in alternating-bit validation. A fragment of the set of states obtained in the protocol validation is given in Fig. 2. The state has there the following form:
Number of states +W11 – W12 – W21 – W22 – W31–
Bit1 – Bit2 – Bit3.

Conclusions. The aggregative specifications analysis method proposed in the paper does not require use or construction of special programing means, implementing the process of the analysis of aggregative models.The problem of aggregative models analysis is reduced to constructing a set of axioms and checking their consistency by the resolution method using the logical programming language PROLOG.

REFERENCES

Bochmann, G.V. (1987). Usage of protocol development tools : the results of a survey. (invited paper). In: *7-th IFIP Symposium on Protocol Specification, Testing and Verification.* Zurich.

Gorelik, Y., and H. Pranevitchus (1985). Automated implementation system for simulation models. In: *Teorija i Modelirovanie Slozhnych Sistem,* Acad. Sci.USSR, Moscow. pp. 83–92. (in Russian).

Pranevitchus, H. (1982). ***Models and Methods for Computer System Investigation.*** Mokslas, Vilnius. 228pp. (in Russian).

Pranevitchus, H., and A. Chmieliauskas (1983). *Correctness Proof and Performance Predication of Protocols Using Aggregate Approach and Control Sequence Method.* Acad. Sci. USSR, Moscow. 32pp. (in Russian).

Pranevitchus, H. and V. Janilionis (1985). Aggregative simulation sistem (SIMAS). In: *Perspectivy Razvitija Vychisliteljnych Sistem,* RPI, Riga. pp. 147–149. (in Russian).

Pranevitchus, H., A. Chmieliauskas, and V. Pilkauskas (1985). Protocol symulation and verification in PRANAS. In: *In Seti i Komutacija Paketov,* ESTI, Riga. pp. 209–213. (in Russian).

Pranevitchus, H., and A. Panevėžys (1988). Proof-of-corretness technique for Aggregative Models of protocols. In:*IFAC/ IMACS International Symposium DIS'88,* Varna. pp. 100–105.

Sidhu, D.P., and C.S. Crall (1988). Executable Logic Specifications for Protocol Servise Interfaces. *Transactions on Software Engineering,* **14**(1), 98–121.

Received February 1991

H. Pranevitchus received the Degrees of Candidate of Technical Scienses at the Kaunas Politechnic Institute, Kaunas, Lithuania and Computer Science Institute of Latvian Academy of Scienses, Riga, Latvija, in 1970 and 1984, respectively. He is currently Professor and head the Departament of Control Systems,Kaunas University of Technology. His research interests include simulation of complex systems and specification, validation, testing and simulation of computer networks protocols.

R. Čeponytė graduated Kaunas Politechnic Institute on 1981. She is currently a post-graduate student at Kaunas University of Technology. She investigates computer network protocol specification and validation.