Modeling Spiking Neural Terminals in DEVS

Yuri Boiko and Gabriel Wainer
Carleton University
1125 Colonel By Drive,
Ottawa, ON, K1S 5B6 CANADA
yuri.boiko@rocketmail.com , gwainer@sce.carleton.ca

Keywords: Discrete event simulation, DEVS, spiking neuron, neural network, atomic model, coupled model.

Abstract

We introduce the simulation of a spiking neural terminal using the CD++ toolkit. Operation of the neuron spike sequences are split between two channels – one for initiating and another one for terminating spikes. The firing condition for the spiking neuron is reached when two rectangular responses, one for the initiating spike and another one for terminating spike, overlap in time domain. A Coupled model of the spiking neuron consists of two atomic models, the timer and the controller, which ensure detection of the spikes in time domain and reaction of the neuron.

1. INTRODUCTION

Modeling of the various elements of the Brain Machine, particularly those involving spiking neurons, is currently in the focus of the simulation and design research communities [1 –6]. The expected technological impact of the progress in the area, as well as due to its benefits, can influence design optimization and operational efficiency of complex systems.

DEVS (Discrete Event System Specification) recently gained recognition for its usefulness in modeling various systems of artificial and/or natural descent [7, 8]. DEVS atomic models are used as building blocks of more complex coupled models, which constitute next level in the hierarchy of model complexity. In turn, coupled models can be used as building blocks for the next hierarchical levels, thus opening way of creating models of any desired level of sophistication. The discrete-event representation, in which only meaningful events are accounted for in simulation, allows retaining speed of the simulation even of highly complex models. Herewith we show how the CD++ toolkit was used for programming the hierarchy of models, which allow constructing Spiking Neural Terminal – the essential part of various devices of the Brain Machine [9 - 11].

2. SPIKING NEURON TERMINAL MODEL

Each spiking neuron was represented as a terminal consisting of two parallel routing lines connecting two nodes with rectangular response function, as shown in Fig.1. One routing line (1) is for the initiating spikes (odd spikes) while the other one (2) is for terminating spikes (even spikes). Nodes (3) and (4) are to generate rectangular response for the incoming spike as well as node (3) of the initiating spike is introducing time delay **$\Delta\tau$** to ensure that only for the spikes with predetermined time interval between initiating and terminating spikes the rectangular pulses overlap. The overlapping of the rectangular pulses (which are the node's responses to the incoming spikes) ensures that the peak amplitude of the signal at the output node reaches threshold value or neuron firing condition, which makes the output node to produce "1". Alternatively, when pulses do not overlap in time, the threshold condition is not met and the output node of the neuron produces "0" (non-firing output).

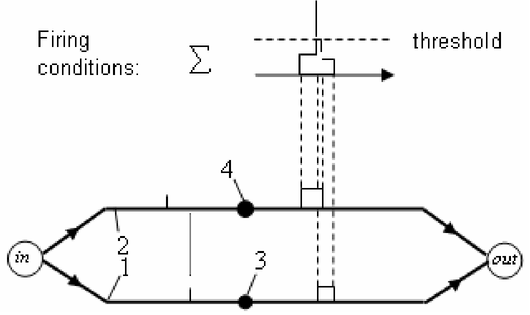
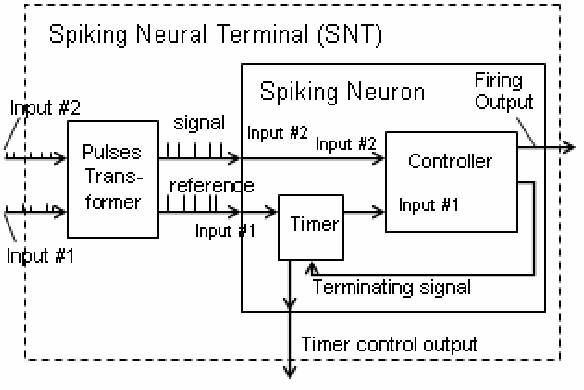


Figure 1. Schematic of the Spiking Neural Terminal.

Fig. 2 shows the coupled model of the spiking neuron terminal model, which includes a timer, and a controller to generate proper outputs depending on the time interval between initiating and terminating spikes.

The model uses two inputs: the reference (a time reference point to measure the delay), and the signal input. The Neuron will fire iif the signal pulse is following the reference within the required time frame (here, $5 \text{ ms} \leq t \leq 8 \text{ ms}$). Other pulses will be discarded. The SNT coupled model includes a Spike/Pulse Transformer (amplifies the incoming sequences of pulses to both inputs) and the Spiking Neuron just described. The behavior of Spike Transformer is to adjust amplitude of the incoming spikes to the required level, satisfying sensitivity of the circuits. As a result the two series of the spikes with the same amplitude (1 unit), but separated by various time intervals. Identification of qualified spike sequences is done by the neuron via interaction of the internal Timer with the output Controller.



```
[top]
components : Transformer Neuron
in : in_1 in_2
out : terminal_output control_output
Link : in_1 in_1@Transformer
Link : in_2 in_2@Transformer
Link : out_1@Transformer neuron_on@Neuron
Link : out_2@Transformer neuron_off@Neuron
Link : neuron_out@Neuron terminal_output
Link : clk_control@Neuron control_output
[Transformer]
components : amp_1@Amp_1 amp_2@Amp_2
in : in_1  in_2
out : out_1 out_2
Link : in_1 in_1@amp_1
Link : out_1@amp_1 out_1
Link : in_2 in_2@amp_2
Link : out_2@amp_2 out_2
[amp_1]
cycle : 00:00:00:000
[amp_2]
cycle : 00:00:00:000
[Neuron]
components : timer@Timer controller@Controller
in : neuron_on neuron_off
out : neuron_out clk_control
Link : neuron_on m_inTurnOn@timer
Link : out_clk@timer clk_control
Link : out_count@timer m_inCount@controller
Link : neuron_off m_in@controller
Link : m_outFire@controller neuron_out
Link : m_outOff@controller m_inTurnOff@timer
[timer]
in : m_inTurnOn m_inTurnOff
out : out_clk out_count
[controller]
in : m_in m_inCount
out : m_outFire m_outOff
```

Figure 2. Coupled model of Spiking Neural Terminal.

3. SUBMODEL DEFINITION AND TESTING

The different atomic and coupled models were tested using the individually, as we will discuss in this section. The first model to test is the Amplifier, considering the sequence of the input spikes and their transformation into the sequence of the spikes with larger amplitude. Table 1, shows a testing scenario for this model

Table 1. Test data verifying operation of the Amplifier

Initiating Events	Outputs
00:000 in_1 0.1	00:000 out_1 1
00:006 in_1 0.2	00:006 out_1 2
00:007 in_1 0.2	00:007 out_1 2
00:008 in_1 0.2	00:008 out_1 2
00:009 in_1 0.2	00:009 out_1 2
00:010 in_1 0.2	00:010 out_1 2
00:011 in_1 0.2	00:011 out_1 2
00:012 in_1 0.2	00:012 out_1 2
00:014 in_1 0.1	00:014 out_1 1

Table 2. Test data for atomic model Timer

RULE	Input Events	Outputs
i	00:000 m_inTurnOn 1.0 --------------------- 00:025 m_inTurnOn 1.0	00:001 out_clk 1 00:001 out_count 1 ------------------ 00:026 out_clk 1 00:026 out_count 1
ii	00:010 m_inTurnOff 1.0 00:040 m_inTurnOff 1.0	00:009 out_count 9 and no output the next cycle ------------------ 00:039 out_count 14 and no output the next cycle
iii	00:025 m_inTurnOn 1.0	00:026 out_clk 1 00:026 out_count 1 00:027 out_clk -1 00:027 out_count 2 00:028 out_clk 1 00:028 out_count 3 00:029 out_clk -1 00:029 out_count 4
iv	00:035 m_inTurnOn 1.0	00:034 out_clk 1 00:034 out_count 9 00:035 out_clk -1 00:035 out_count 10 00:036 out_clk 1 00:036 out_count 11
v	00:010 m_inTurnOff 1.0 00:020 m_inTurnOff 1.0 00:025 m_inTurnOn 1.0	00:009 out_clk 1 00:009 out_count 9 00:026 out_clk 1 00:026 out_count 1

The Timer model is activated by the input of the reference pulse (spike) via input1= m_inTurnOn and terminated by the stop signal (of 1) via input2= m_inTurnOff.

The following properties need to be verified in the Timer's test (rules (i)-(v)):

(i) internal counting in the timer starts exactly on the entry of the reference signal (with no time delay on that);

(ii) internal counting terminates at exactly same time as stop signal arrives;

(iii) in the active state intervals between the state change of the parameter ***clk*** ={(+1); (-1)} are exactly equal to the value of ***cycleTime*** parameter, i.e. 1 msec.;

(iv) reference spikes arriving in the active state do not interrupt the counting of the ***clk*** cycles;

(v) stop signal arriving at the passive state does not activate the ***clk*** counter of the Timer.

The following sequence table shows the resulting outputs for this model. We can see that the internal cycles counting in the Timer start exactly on the entry of the reference signal (with no time delay on that), because the very key points of Timer activation produce values out_***clk***=1 and out_count=1 the first output after the occurrence of activation entry. This is what is expected and confirming rule (i), because in passive state the respective values of the variables are ***clk***=-1 and ***count***=0. The next rule (ii) is also confirmed that internal counting terminates at exactly same time as stop signal arrives, as seen from the fact that termination signal "`m_inTurnOff 1.0`" prevents output immediately, as there is no output at and after the timing of termination signal. The third rule (iii - active state intervals between the state change of the parameter *clk* ={(+1); (-1)} are exactly equal to the value of ***cycleTime*** parameter) as illustrated in Table 2. The reference spike (`00:035 m_inTurnOn 1.0`) arriving in the active state is seen as not interrupting the counting of the ***clk*** cycles, thus confirming rule (iv) (Table 2, (iv)). And finally, the stop signal (`00:020 m_inTurnOff 1.0`) arriving at the passive state of the Timer does not activate the ***clk*** counter of the Timer, as it is seen in the non-interrupted inactivity of the passive state until the next activating influence by arriving reference spike, as required by the rule (v) and gathered in the Table 2, (v).

Similar tests were carried out for the Controller, the Spiking Neuron coupled model, and the Pulses Transformer coupled model.

The Controller is activated by the input of the signal spike via input2 and self-terminates after taking the first available count at the reference input (input1) and producing the output events {1;0} at the output1 and {1} at the output2.

The Controller's properties needed to be verified are (vi-x):

(vi) that the Controller is producing outputs only when spike via input2 is followed by the signal intake via input1 (and not otherwise, as states the next criteria vii);

(vii) signals to input1 are ignored unless it was preceded by the signal at input2;

(viii) firing output of 1 is resulting only when the value the count from the input1 satisfies the firing condition (i.e. 5 msec ≤ count ≤ 8 msec);

(ix) when firing condition is not met, the output at output1 is 0, even though other conditions of Controller activation are met;

(x) firing condition holds on the borders of the interval (i.e. upper and lower limits of the interval are still valid firing conditions).

The confirmation that as required by the rule (vi) the Controller is producing outputs only when spike via input2= m_in is followed by the signal intake via input1= m_inCount is seen in the fact that all combinations
`00:00:00:00t m_in 1`
`00:00:00:00(t+1) m_inCount a`
do produce the response at the both output ports (even when non-firing signal of 0 occurs at the `output1= m_outfire`. This is summarized in the Table 3,(vi). Combination
`00:00:00:00t m_in 1`
`00:00:00:00t m_inCount a`
does the same as the previous one, which means that time shift between the two input signals can be anything within interval [0 – cycleTime] (also included in the Table 3,(vi)).

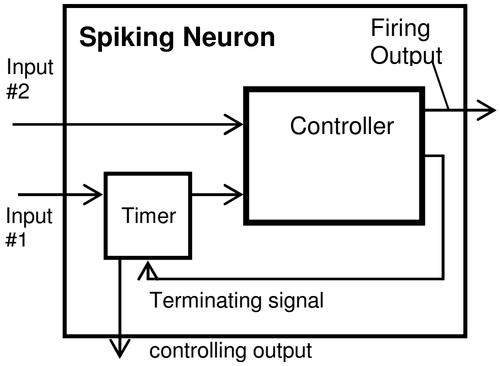
Table 3. Test data confirming the rules (vi)-(x) for the atomic model of Controller.

R U L E	Initiating Events (the cause)	Output result, confirming the rule
1	2	3
vi	00:00:00:002 m_in 1 00:00:00:003 m_inCount 6 00:00:00:007 m_in 1 00:00:00:008 m_inCount 2 ----------------------- 00:00:00:012 m_in 1 00:00:00:012 m_inCount 7	00:00:00:003 m_outoff 1 00:00:00:003 m_outfire 1 00:00:00:008 m_outoff 1 00:00:00:008 m_outfire 0 ----------------------- 00:00:00:012 m_outoff 1 00:00:00:012 m_outfire 1
vii	00:00:00:015 m_inCount 7 and 00:00:00:020 m_in 1 00:00:00:021 m_inCount 9	Ignored, no outputs ------------------------ 00:00:00:021 m_outoff 1 00:00:00:021 m_outfire 0
viii	00:00:00:002 m_in 1 00:00:00:003 m_inCount 6 00:00:00:012 m_in 1 00:00:00:012 m_inCount 7	00:00:00:003 m_outoff 1 00:00:00:003 m_outfire 1 00:00:00:012 m_outoff 1 00:00:00:012 m_outfire 1
ix	00:00:00:007 m_in 1 00:00:00:008 m_inCount 2 00:00:00:020 m_in 1 00:00:00:021 m_inCount 9	00:00:00:008 m_outoff 1 00:00:00:008 m_outfire 0 00:00:00:021 m_outoff 1 00:00:00:021 m_outfire 0
x	00:00:00:030 m_in 1 00:00:00:031 m_inCount 5 00:00:00:040 m_in 1 00:00:00:041 m_inCount 8	00:00:00:031 m_outoff 1 00:00:00:031 m_outfire 1 00:00:00:041 m_outoff 1 00:00:00:041 m_outfire 1

Verification of the Controller's feature (rule (vii)) that signals to input1= m_inCount are ignored unless it were preceded by the signal at input2= m_in is seen in the fact that the singular input of
`00:00:00:015 m_inCount 7`
unaccompanied by the preceding or simultaneous input to the port **`m_in`** is ignored, while others with accompanying pulse are not, (see Table 3,(vii)). Spiking neuron firing output of 1 is resulting only when the value the count from the input1= m_inCount satisfies the firing condition (i.e. 5 msec ≤ count ≤ 8 msec), as it is seen in the fact of the rule (vii) requires and the Table 3,(vii) summarizes. Outside of the firing timing interval (i.e. 5 msec ≤ count ≤ 8 msec), for example as in the events below the lower limit, i.e. too early spike (`00:00:00:008 m_inCount 2`), as well as above it, as too late spike of (`00:00:00:021 m_inCount 9`) – both produce non-firing output of 0, verifying the rule (ix) prescription (see Table 3,(ix)). The borders of the interval are also verified by events (`00:00:00:031 m_inCount 5`) and

(`00:00:00:041 m_inCount 8`) producing firing output 1, as the rule (x) requires (see Table 3,(x)).

The coupled model of Spiking Neuron consists of two components or sub-models: the Timer and the Controller (Fig.3). The Coupled model validation requires verification of the consistency in the interaction between the sub-models under various relations in the input combinations. In the tests of the Atomic models such interaction was substituted by the external events listed in the ***.ev** files, separate for each sub-model.



```
[top]
components : timer@Timer controller@Controller
in : neuron_on neuron_off
out : neuron_out clk_control
Link : neuron_on m_inTurnOn@timer
Link : out_clk@timer clk_control
Link : out_count@timer m_inCount@controller
Link : neuron_off m_in@controller
Link : m_outFire@controller neuron_out
Link : m_outOff@controller m_inTurnOff@timer
```

Figure 3. External connections of the coupled model of Spiking Neuron (inserted is it's CD++ code).

Now, the external events will be supplied to the Coupled Model, and the consistency in the internal interactions will be estimated based on the resulting overall output.

For validation purposes in addition to the firing output, which represents the functionality of the model, introduced is one controlling output coming directly from the Timer and which allows following the synchronous change of state of both sub-models during the coupled model operation.

The file ***neuron.ev*** was created as shown in left column of Table 4, which together with sequential neuron_on and neuron_off spikes includes multiple sequential repetition of either the neuron_off spikes as well as neuron_on spikes to verify that such sequences do not disrupt neither fully passive state of the Spiking Neuron as a whole as well as activated Timer in combination with passive state of the Controller. Additionally, various intervals are tested between neuron_on and neuron_off spikes to verify firing interval conditions within the interval itself, as well as on its borders and on outside of the interval, shown in the right column of the Table 4.

Table 4. Test data confirming the rules (xi)-(xv) for the coupled model of the Spiking Neuron, comprising Timer and Controller as components

rule	Initiating Events (the cause)	Output result, confirming the rule
xi	00:00:00:000 neuron_on 1 00:00:00:006 neuron_off 1	00:00:00:001 clk_control 1 00:00:00:002 clk_control -1 00:00:00:003 clk_control 1 00:00:00:004 clk_control -1 00:00:00:005 clk_control 1 00:00:00:006 clk_control -1 00:00:00:006 neuron_out 1
xii	00:00:00:014 neuron_on 1 00:00:00:016 neuron_off 1 ------------------------- 00:00:00:019 neuron_on 1 00:00:00:029 neuron_off 1	00:00:00:015 clk_control 1 00:00:00:016 clk_control -1 00:00:00:016 **neuron_out 0** ------------------------- 00:00:00:020 clk_control 1 00:00:00:021 clk_control -1 00:00:00:022 clk_control 1 00:00:00:023 clk_control -1 00:00:00:024 clk_control 1 00:00:00:025 clk_control -1 00:00:00:026 clk_control 1 00:00:00:027 clk_control -1 00:00:00:028 clk_control 1 00:00:00:029 clk_control -1 00:00:00:029 **neuron_out 0**
xiii	00:00:00:031 neuron_on 1 00:00:00:036 neuron_off 1 -------------------------- 00:00:00:038 neuron_on 1 00:00:00:046 neuron_off 1	00:00:00:032 clk_control 1 00:00:00:033 clk_control -1 00:00:00:034 clk_control 1 00:00:00:035 clk_control -1 00:00:00:036 clk_control 1 00:00:00:036 **neuron_out 1** -------------------------- 00:00:00:039 clk_control 1 00:00:00:040 clk_control -1 00:00:00:041 clk_control 1 00:00:00:042 clk_control -1 00:00:00:043 clk_control 1 00:00:00:044 clk_control -1 00:00:00:045 clk_control 1 00:00:00:046 clk_control -1 00:00:00:046 **neuron_out 1**
xiv	00:00:00:006 neuron_off 1 00:00:00:007 neuron_off 1 00:00:00:008 neuron_off 1 00:00:00:009 neuron_off 1 00:00:00:010 neuron_off 1 00:00:00:011 neuron_off 1 00:00:00:012 neuron_off 1 00:00:00:014 neuron_on 1	00:00:00:006 clk_control -1 00:00:00:006 neuron_out 1 00:00:00:015 clk_control 1
xv	00:00:00:048 neuron_on 1 00:00:00:049 neuron_on 1 00:00:00:050 neuron_on 1 00:00:00:051 neuron_on 1 00:00:00:052 neuron_on 1 00:00:00:053 neuron_on 1 00:00:00:054 neuron_off 1	00:00:00:049 clk_control 1 00:00:00:050 clk_control -1 00:00:00:051 clk_control 1 00:00:00:052 clk_control -1 00:00:00:053 clk_control 1 00:00:00:054 clk_control -1 00:00:00:054 **neuron_out 1**

The rules to be verified for the coupled model of Spiking Neuron are (xi-xv):

(xi) within firing interval of the qualified spike sequences the Spiking Neuron produces firing output of 1;

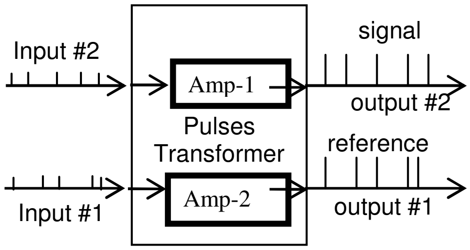
(xii) outside of the firing interval of the qualified spiking sequences the Spiking Neuron does produce non-firing output of 0;

(xiii) on the borders of the firing interval of the qualified spiking sequences the Spiking Neuron does produce firing output of 1;

(xiv) repetitions of signal spike sequences do not disturb passive state of the whole Neuron unless there if reference spike;

(xv) repetitive reference spikes do not disturb combination of active Timer with passive Controller, which represent the waiting state of the Spiking Neuron for the signal spike to come; importantly, only first reference spike counts, and the rest are ignored until the input of the signal spike.

The rule (xi) is verified by sequences of the qualified spikes within the firing interval of the Spiking Neuron to produce firing output of 1. Outside of the firing interval even for qualified spiking sequences the Spiking Neuron produces non-firing output of 0, thus confirming rule (xii). For the upper and lower limits of the firing interval the qualified spiking sequences result in the firing output of 1 of the Spiking Neuron, thus confirming validity of the borders of the firing interval of rule (xiii). Repetitions of signal spikes in a sequences shown not disturb the passive state of the whole Neuron (i.e. those are ignored, producing no *clk*-outputs) unless the reference spike comes in, which verifies rule (xiv). Similarly repetitive reference spikes do not disturb operation of active Timer under passive Controller, in which case the waiting state of the Spiking Neuron is maintained until arrival of the signal spike. In this case only first reference spike produces effect (activates the Timer) while others are disregarded until obtaining the input of the signal spike, thus verifying the rule (xv).



```
[top]
components : amp_1@Amp_1 amp_2@Amp_2
in : in_1 in_2
out : out_1 out_2
Link : in_1 in_1@amp_1
Link : out_1@amp_1 out_1
Link : in_2 in_2@amp_2
Link : out_2@amp_2 out_2
```

Figure 4. Pulses Transformer Coupled Model.

The coupled model of Pulses Transformer includes two independent and identical atomic models of Pulses Amplifiers (see Fig.4), obeying the rule (xvi):

(xvi) the two parallel series of spikes are amplified independently, i.e. with different amplification coefficient, routed to different outputs with preserved time differences.

The tests for this coupled model are shown in Table5. It is seen that (1) difference in the amplitude between two series is eliminated, (2) connections between in_1 to out_1 and in_2 to out_2 are kept, as well as (3) time labels are preserved. The spike (`00:000 out_2 0`) at the output 2 is 0, because there was no actually input for in_2 for the starting point (left column).

Table 5. Test data for the Pulses Transformer

Input Events	Outputs
`00:000 in_1 0.1`	`00:000 out_1 1`
	`00:000 out_2 0`
`00:006 in_2 0.2`	`00:006 out_2 1`
`00:007 in_2 0.2`	`00:007 out_2 1`
`00:008 in_2 0.2`	`00:008 out_2 1`
`00:009 in_2 0.2`	`00:009 out_2 1`
`00:010 in_2 0.2`	`00:010 out_2 1`
`00:011 in_2 0.2`	`00:011 out_2 1`
`00:012 in_2 0.2`	`00:012 out_2 1`
`00:014 in_1 0.1`	`00:014 out_1 1`

4. SPIKING NEURAL TERMINAL SIMULATOR

The coupled model of Spiking Neural Terminal (SNT) was schematically presented in Fig.2, and it is composed of two coupled sub-models: (1) Pulses Transformer and (2) Spiking Neuron.

Functionally, the top model of SNT appears to be very similar to that of the coupled model of Spiking Neuron itself with only one addition – the spike source represented by the list of the spiking events undergoes amplification by Pulses Transformer prior to transfer to the inputs of Spiking Neuron. Therefore, the major additional rule, we need to be verified in the top-model SNT is that the rules for the coupled model of Spiking Neuron are preserved for the SNT model, but complimented with the rules of the coupled model of Pulses Transformer.

This means that offering to the SNT model an event list same as to the Neuron model, but with reduced amplitude (to enable Pulses Transformer to compensate for the amplitude implementing rule) has to result in the identical output for both models – SNT and Neuron's one

This is illustrated in Table 6, where first row (titled "INPUTS") shows similarity in the offered events lists for the Neuron (on the left) and for the SNT with amplitude being the only difference. Second row (titled "OUTPUTS") compares the outputs for both models, which are seen to be identical. This verifies the validity of the coupled model under consideration – Spiking Neuron Terminal Simulator. It is seen from the Table 6 that the initial differences in the event list of the input spikes produces the logically sound differences in the output of the two models until the time slot of 00:015, starting from which the preserved similarity in the input events results in the identical end output for both models. By this the validity of the rule (xvii) is confirmed by sample events verification, thus validating SNT model in DEVS environment.

Table 6. Comparative test of SNT and Spiking Neuron Coupled Models

	Spiking Neuron Model	SNT Model
INPUTS		`00:000 in_2 0.2`
	`00:000 neuronOn 1`	`00:000 in_1 0.1`
		`00:001 in_1 0.1`
	`00:006 neuronOff 1`	`00:006 in_2 0.2`
	`00:007 neuronOff 1`	`00:007 in_2 0.2`
	`00:008 neuronOff 1`	`00:008 in_2 0.2`
	`00:009 neuronOff 1`	`00:009 in_2 0.2`
	`00:010 neuronOff 1`	`00:010 in_2 0.2`
	`00:011 neuronOff 1`	`00:011 in_2 0.2`
	`00:012 neuronOff 1`	`00:012 in_2 0.2`
	`00:014 neuronOn 1`	`00:014 in_1 0.1`
	`00:016 neuronOff 1`	`00:016 in_2 0.2`
	`00:017 neuronOff 1`	`00:017 in_2 0.2`
	`00:018 neuronOff 1`	`00:018 in_2 0.2`
	...	...
	`00:049 neuronOn 1`	`00:049 in_1 0.1`
	`00:050 neuronOn 1`	`00:050 in_1 0.1`
	`00:051 neuronOn 1`	`00:051 in_1 0.1`
	`00:052 neuronOn 1`	`00:052 in_1 0.1`
	`00:053 neuronOn 1`	`00:053 in_1 0.1`
	`00:054 neuronOff 1`	`00:054 in_2 0.2`
OUTPUTS	`00:001 clk_ctl 1`	`00:001 ctlOutput 1`
	`00:002 clk_ctl -1`	`00:001 terminalOutput 0`
	`00:003 clk_ctl 1`	
	`00:004 clk_ctl -1`	
	`00:005 clk_ctl 1`	
	`00:006 clk_ctl -1`	
	`00:006 neuronOut 1`	`00:013 terminalOutput 0`
	`00:015 clk_ctl 1`	`00:015 ctlOutput 1`
	`00:016 clk_ctl -1`	`00:016 ctlOutput -1`
	`00:016 neuronOut 0`	`00:016 terminalOutput 0`
	`00:020 clk_ctl 1`	`00:020 ctlOutput 1`
	`00:021 clk_ctl -1`	`00:021 ctlOutput -1`
	`00:022 clk_ctl 1`	`00:022 ctlOutput 1`
	...	...
	`00:028 clk_ctl 1`	`00:028 ctlOutput 1`
	`00:029 clk_ctl -1`	`00:029 ctlOutput -1`
	`00:029 neuronOut 0`	`00:029 terminalOutput 0`
	`00:032 clk_ctl 1`	`00:032 ctlOutput 1`
	`00:033 clk_ctl -1`	`00:033 ctlOutput -1`
	`00:034 clk_ctl 1`	`00:034 ctlOutput 1`
	`00:035 clk_ctl -1`	`00:035 ctlOutput -1`
	`00:036 clk_ctl 1`	`00:036 ctlOutput 1`
	`00:036 neuronOut 1`	`00:036 terminalOutput 1`
	`00:039 clk_ctl 1`	`00:039 ctlOutput 1`
	`00:040 clk_ctl -1`	`00:040 ctlOutput -1`
	`00:041 clk_ctl 1`	`00:041 ctlOutput 1`
	`00:042 clk_ctl -1`	`00:042 ctlOutput -1`
	`00:043 clk_ctl 1`	`00:043 ctlOutput 1`
	`00:044 clk_ctl -1`	`00:044 ctlOutput -1`
	`00:045 clk_ctl 1`	`00:045 ctlOutput 1`
	`00:046 clk_ctl -1`	`00:046 ctlOutput -1`
	`00:046 neuronOut 1`	`00:046 terminalOutput 1`
	`00:049 clk_ctl 1`	`00:049 ctlOutput 1`
	`00:050 clk_ctl -1`	`00:050 ctlOutput -1`
	`00:051 clk_ctl 1`	`00:051 ctlOutput 1`
	`00:052 clk_ctl -1`	`00:052 ctlOutput -1`
	`00:053 clk_ctl 1`	`00:053 ctlOutput 1`
	`00:054 clk_ctl -1`	`00:054 ctlOutput -1`
	`00:054 neuronOut 1`	`00:054 terminalOutput 1`

3. CONCLUSIONS

The following conclusions can be drawn from the above:

1. CD++ toolkit is demonstrated as a suitable environment for simulation of the Spiking Neural Terminal under DEVS formalism.
2. Sub-models of timer and controller can successfully implement rectangular response function of the spiking neuron.
3. The model of the Spiking Neural Terminal capable of detecting pre-programmed spike sequences and based on spiking neurons with rectangular response function is successfully implemented and validated in DEVS formal definitions.
4. Hierarchy of the atomic models of amplifier, timer and controller, as well as coupled models of Pulses Transformer and Spiking Neuron comprising the top model have been successfully validated together with the top model of Spiking Neural Terminal.

References:

[1] Zeigler, B.P., The brain-machine disanalogy revisited, *BioSystems,* Vol. 64, pp. 127-140. (2002).

[2] Buller A (2003a) CAM-Brain Machine and Pulsed Para-Neural Networks (PPNN): Toward a hardware forfuture robotic on-board brains, *Proceedings of the Eighth International Symposium on Artificial Life and Robotics (AROB 8th '03), January 24-26, 2003, Beppu, Oita, Japan*, 490-493.

[3] Eeckhaut H, Van Campenhout J., (2003) "Handcrafting Pulsed Neural Networks for the CAM-Brain Machine",- *Proceedings of the Eighth International Symposium on Artificial Life and Robotics (AROB 8th '02),January 24-26, 2002, Beppu, Oita, Japan*, 494-501.

[4] M. Conrad, "The brain-machine disanalogy"- Biosystems, 1989 - vol.22, no.3, pp197-213, Elsevier, 1989.

[5] Michael Korkin1, Norberto Eiji Nawa, Hugo de Garis, A "Spike Interval Information Coding" Representation for ATR's CAM-Brain Machine (CBM) Volume 1478 (1998).

[6] Obeid, I. Wolf, P.D, "Evaluation of spike-detection algorithms fora brain-machine interface application",- Biomedical Engineering, IEEE Transactions on, Volume 51, Issue 6, page(s) 905- 911, June 2004.

[7] Zeigler, B.P., DEVS Component-Based M&S Framework: An Introduction , Topics Seminar: Object Oriented Simulation-Discrete Event Modeling Spring 2007.

[8] G. Wainer. "CD++: a toolkit to define discrete-event models". G. Wainer. In *Software, Practice and Experience*. Wiley. Vol. 32, No.3. November 2002. pp. 1261-1306

[9] R Mayrhofer, M Affenzeller, H Prahofer, G Hofer, A., "DEVS Simulation of Spiking Neural Networks", - Proceedings of Cybernetics and Systems (EMCSR), 2002.

[10] Thomas Natschläger, Wolfgang Maass, "Fast Analog Computation in Networks of Spiking Neurons."- (1999) http://www.cis.tu-graz.ac.at/igi/maass/src-esann99.ps.gz

[11] Wolfgang Maass, "Networks of Spiking Neurons: The Third Generation of Neural Network Models", - (1996) http://www.cis.tu-graz.ac.at/igi/maass/85j.ps.gz