Modeling Attention Switching in Resource-constrained Complex Intelligent Dynamical Systems (RCIDS)

Saurabh Mittal
Dunip Technologies, Tempe, AZ
smittal@duniptech.com

Bernard P. Zeigler
RTSync Corp. Rockville, MD
zeigler@rtsync.com

Keywords: DEVS, Attention switching, complex intelligent dynamical systems, bottom-up phenomenon, top-down phenomenon, sampling, quantization, rate estimator, RCIDS

Abstract

Sustainable natural systems require energy. This requirement of energy is proportional to the activity they manifest. In a scalable self-similar artificial system, information gateways at every level must limit the information/activity to-from their subsystems based on computational algorithms. We discuss some of the implemented algorithms in these gateways for focusing attention to the most active component. We validate that attention switching is an emergent property of such a system and happens when there is goal-directed behavior within the system or an agent. We describe the structural elements that are needed to deal with the bottom-up and top-down phenomenon and model the behavior using DEVS-based hierarchical system capable of focusing attention. We define resource-constrained complex intelligent dynamical system (RCIDS) and summarize various application areas that can leverage these concepts.

1. INTRODUCTION

Complex natural systems (CNS) bear self-similar properties and various studies show that they are self-similar [1]. Nature supports hierarchical architecture. At each level of the hierarchy in CNS, the information is processed and filtered and then transferred to the higher level. Pinker [9] provides the fundamental guidelines of an artificial system that tries to imitate natural systems.

"Any intelligent agent incarnated in matter, working in real-time and subset to the laws of thermodynamics must be restricted in its access to information. Only the information that is relevant should be allowed in."

When scalability and measurement of intelligence is viewed from the Computer Science perspective, then intelligent processing can be viewed as a subset of universal computation. So, intelligence is measured by standard computational complexity. Intelligent systems are organization of algorithms operating in n-tier hierarchy, each of which may become a bottleneck for growth. The notion of intelligence is multi-faceted, subjective and adding "intelligence" in a system always has a cost, whether computational time, energy, resources or knowledge. Given the apparently complexity of such a system at both the design/compile time or at run-time, how does one focus its attention at a specific feature of such system? Attention is defined as the capacity to direct one's resources (or mind in psychological terms) preferentially to an object from a set of complex stimuli, thereby reducing their footprint. While attention switching occurs naturally, the *focus* of attention is a deliberate, top-down phenomenon guided by a goal-directed behavior.

An Algorithm in such a system is considered practical if it completes in polynomial execution time proportional to the input data. Likewise, a problem is tractable if it has a polynomial algorithm solution. However, there are intractable problems also known as NP complete problems that have non-deterministic polynomial execution time. Tsotsos et.al showed that the problem of searching a visual space is one such intractable problem when the targets are not known in advance but becomes a tractable one if a target is given [2-6]. In other words, the top-down behavior of knowing a target in advance, helps to "focus" attention as it switches from one object to another object.

The human cognitive ability to attend has been widely researched in cognitive and perceptual psychology, neurophysiology and in computational systems and the core issue has been of *Information Reduction* [3]. This capacity to attend has been computationally implemented as a search-limiting heuristic in early AI literature. Various cognitive architectures implemented information reduction in different ways, mostly to limit the "working memory" component but still fail to explicitly discuss human capacity, bottlenecks and resource limits [3]. Computational models of Attention [3] can be specialized into four hypotheses:

1. *Salience Map*: Problem is represented as a set of feature maps of various stimuli. Saliency map implements a Winner Take All (WTA) algorithm that combines information from various feature maps resulting into one salient outcome and then inhibiting it so that the next salient feature gains attention.
2. *Temporal Tagging*: Attentional mechanism binds all those neurons whose activity relates to the relevant features of a single object, which form a transient short-term memory. The neuronal system is modeled as a dynamical system.
3. *Emergent Attention*: A property of large assemblies of neurons engaged in competitive interactions and

selection is the combined result of local dynamics and top-down biases.

4. *Selective Routing*: Feed-forward and feedback networks illuminate overlapping neural paths in presence of localized inputs.

Styles [7] suggested that attentional behavior emerges as a result of complex underlying processing in brain and Shipp's review [8] concludes that this emergent attention is the most likely hypothesis.

An attempt to design such an attention-management architecture is presented in this paper and is based on Mittal's Master's thesis [10]. We define a resource-constrained scalable complex intelligent dynamical system (RCIDS) with the following properties:

1. *Resource constrained environment*: In the modeled connectionist system, network bandwidth, and computational resources available to any sensor are finite. The constraints may take the form of energy, time, knowledge, control, etc. that are available to any processing component
2. *Complex*: Presence of emergent behavior that is irreducible to any specific component in the system. Attention switching is an emergent phenomena.
3. *Intelligent*: The capacity to process sensory input from the environment and act on the sensory input by processing the information to pursue a goal-oriented behavior
4. *Dynamical*: The behavior is temporal in nature. The system has emergent response and stabilization periods.
5. *System*: The model conforms to systems theoretical principles.

The proposed RCIDS architecture is a prototype of an intelligent system capable of focusing attention and directing resources to an area showing high activity. Any natural or artificial system that has finite amount of resources and has a requirement to focus attention/resources to a region of high "activity" can be mapped onto this architecture. Any artificial system is composed of sensors, a data/information processing engine capable of decision making and selective control, and actuators. This work investigates the capability to acknowledge the detected activity, worthy of attention for a given task-at-hand, register it and then releasing commands to the appropriate actuators. Following are some of the enterprise system of systems (SoS) where this system could be put to test:

a) Human decision-making System or Decision aides
b) Autonomous systems & Robotic systems
c) Applications involving Hierarchically distributed Genetic Algorithms
d) Learning Management systems
e) Learning Health systems

In this paper, we show by simulation that attention switching is an emergent property of a resource-constrained system and goal-directed behavior (implemented as various algorithms) results in focusing attention to a component showing a higher activity which in turn uses more energy. Section 2 provides a scientific background behind our approach. Section 3 discusses the model system design. Section 4 presents the model abstractions and theoretical analysis of various sampling algorithms. Section 5 discusses simulation results. Section 6 summarizes the article.

2. APPROACH

In order to display an intelligent reactive time-critical responsible behavior, it is imperative to have a "reasoning" component capable of making selective decisions, knowledge refinements, resource allocations and the involved attention switching. Emergence of hierarchical structures from bottom-up phenomena occurs in natural complex systems and emergence of clusters and hubs appears to be aided by top-down phenomena [1]. Such systems are self-similar or fractal in nature and often studied as complex adaptive systems [11]. When viewed from systems perspective, the *decision* making is goal-oriented and follows the top-down approach while the information flows according to the bottom-up approach.

We make some bold assumptions to limit the scope of our problem and focus only on the essential features of RCIDS. These abstractions are necessary to address the attention focusing problem, independent of the domain knowledge. If applied to a specific domain, each node in such SoS will then subscribe to its utility with respect to the task-at-hand. Following are the assumptions made:

1. The nature of base activity
2. Knowledge/memory structure in any computer-based intelligent system
3. Impact of knowledge activation in attention switching
4. Hierarchical structure of the system
5. Channelization of top-down control with some fundamental selective tuning algorithms at decision making gateways.

With respect to the model and the system architecture, following are some of the concepts that we leverage to build an abstract RCIDS:

2.1.1. Concept of Activity and its relationship with Energy

Energy is the general concept that represents the physical cost of action in the real world. *Information* is the general concept that models how systems decide on, manage, and control their actions [11]. As in Figure 1 a), information and energy are two key concepts whose interaction is well understood in the following common sense manner: On one hand, information processing takes energy, On the other hand, getting that energy requires information processing to find and consume energy-bearing resources. The information processing that a system can do is limited by the energy available to it. However, to increase the amount of energy available to it, a system must use its information processes – but these use some of that energy. A SoS is *sustainable* in the environment if the energy expended by the SoS to meet behavioral requirements is matched by the energy accruing to it by satisfying the requirements.

Activity is a measure of system behavior that allows estimating how much energy a behavior needs to consume. Intuitively, the more active that a component is, the more energy it requires to maintain its activity. Zeigler [12, 13] postulated the *linear allocation strategy*::

$$E_i = aA_i$$

where E_i and A_i are the energy allocated and activity, resp. of the component with subscript i and a is the proportionality factor. Assume that each pattern sensed by the system requires a corresponding distribution of activities among its components to be properly sensed. Then [12] shows that the potential to save total energy using the linear allocation strategy is determined by the *activity disparity*, which is the difference between the maximum and minimum activity of the components, To achieve the linear allocation condition required coordination mechanisms such as the attention focusing architectures to be discussed.

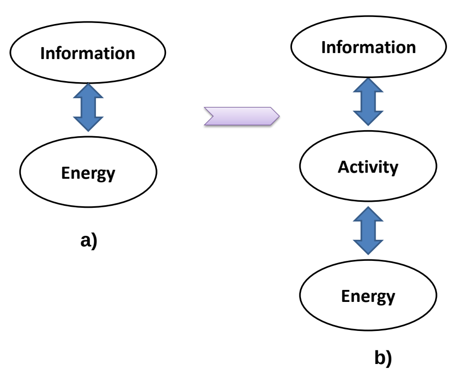

Figure 1. Activity concept linking information and energy

2.1.2. Sampling algorithm at information/knowledge gateways

The Sampling algorithm is an abstraction of a decision making entity at a particular level and has a top-down influence. It works on the Activity concept manifested by any processing agent/sub-system. The underlying logic is an adaptation of Winner Take All (WTA) algorithm [5]. It is executed by a resource-allocation manager or a decision making entity that has a sensitivity threshold (*outQuant*). It is implemented as:

1. Get new activity value (A_t)
2. Calculate the difference with previously stored activity as $dA = A_t - A_{t-1}$
3. Update the sum_t parameter which reflects the total amount of resources being used by all the sensors.
4. Calculate the difference with previously accumulated sum as $dSum = sum_t - sum_{t-1}$
5. if *dSum* > *outQuant* {
    1. $sum_{t-1} = sum_t$
    2. Revise Sampling Rate SR_t based on either of
        a. Normalized-Sum (NS) Rule
        b. Normalized-Max (NM) Rule
        c. Tunable Alfa-Beta (TA) Rule
    3. Send new *sampling rates* to the sensors.
    4. $SR_{t-1} = SR_t$
5. Repeat indefinitely

In this article, we will show the effect of NS, NM, TA sampling algorithms implemented at the resource allocation gateways in the RCIDS model. To minimize energy consumption, the information gateways capable of resource allocation “focus” attention at components displaying abnormal activity.

3. MODEL-SYSTEM DESIGN

We develop an architecture using DEVS systems engineering principles [18]. We develop a hierarchical model analogous to a geographical structure of a country, states, counties, cities and surveillance-areas.

3.1. Model

The system is designed in a top-down manner, where the top level is defined as the geographical region within a country. A country is made up many geographical states. A State is made of many counties. County is made of many cities. City is made of sensory areas, each with a finite area. Each level of hierarchy displays a certain “activity” as is evaluated at each level. Resources and the allocation of resources are proportional to the area. Each sensory-area is modeled as a Cell-grid and composed of various cells (eg. ranging from 50 to 300). Cell stores “resource” (value between 1 and 100) and may turn *active* stochastically, dependent on the resource value. When Cell is active, it utilizes resources and becomes *passive* when resource amount to zero. Activity of a sensor-area is defined as the cumulative number of *active* cells in that area. Sensor-area is provided with Sensor and a Rate Estimator (RE). RE acts as sensor’s decision maker. Sensor has a variable sampling rate, a maximum sampling rate, an activity-detection threshold as managed by RE. City is provided a Sampling Manager (also referred as Resource-allocation Manager [RAM]) that allocates new sampling rate to sensors underneath the hierarchy based on the city’s activity.

An abnormal activity is defined as an activity that has its value greater than the threshold activity as pre-encoded for the sensor. Each city area has a certain activity level and the sensors are tuned to perform at a default level if the city activity level is below a certain threshold. If the activity level increases over this threshold then the sensor also increases its sampling rate so as to process the more information from this increased activity area. The sensor is provided with a capability to modify its threshold so that it can be *sensitive* or *coarse*. It may be done externally by an allocation-manager (e.g., RAM) that maintains balance between the sensor's allocation of resources. This manager also has the current statistics about all the sensors working in different cities within a state (highly simplified situation). The basic job of this manager is to modulate the total resources available with it among the sensors assigned to different cities, implementing a *zero-sum* game. Explicitly, each county will have a manager to supervise and distribute the total resources among the sensors and report the collective usage to its superior level. Though this manager has the capability to set the threshold of the individual sensors it is not mandatory to provide this functionality.

Figure 2 shows a block diagram of the system under discussion containing only one level. It depicts a sensor system (composed of sensor and its RE) interacting with Resource Allocation Manager (RAM). The current *Sampling rate* is communicated to RAM and the sampled data is communicated to the Data-driven Decision Maker (DDM) which processes the data and makes decisions about the sensor activity mode based on the goals of the system at this particular level. The DEVS state machine for Sensor is shown in Figure 3.

3.2. System architecture and design

In reality, the system is actually a multi-level system where the information flows upwards and the allocated resources flow downwards towards the sensors. The communication is done using the network (wireless, wired or radio) channel as the sensors are distributed in real space. A typical hierarchical system is shown in Figure 4 and is called Adaptive Sensor Net (AdsNet) [10]. The hierarchy enforces the information filtering as it travels up the hierarchy. Having described the semantics of the problem-at-hand, the use of System Entity Structure (SES) formalism [14] comes handy at this stage (Figure 4) to describe the structure.

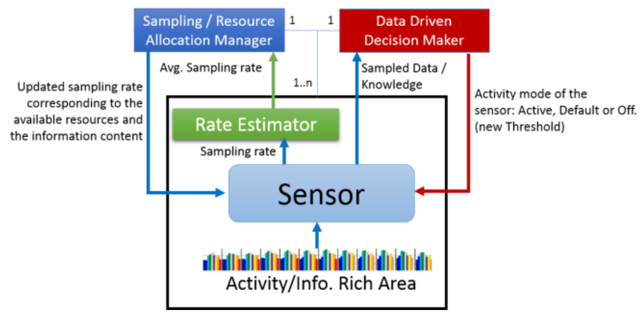

Figure 2. System model for a single level architecture

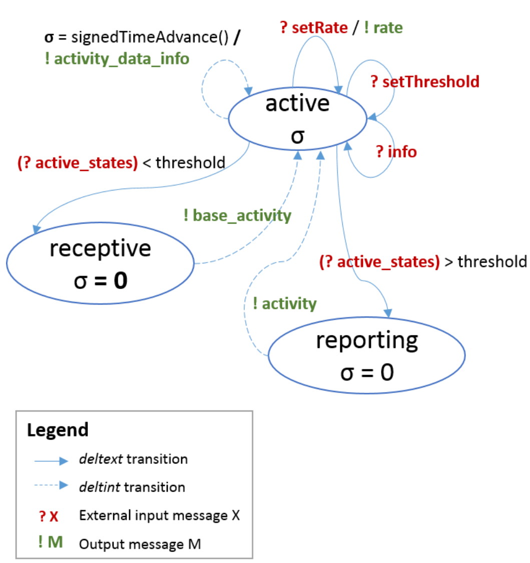

Figure 3. DEVS State machine for Sensor behavior

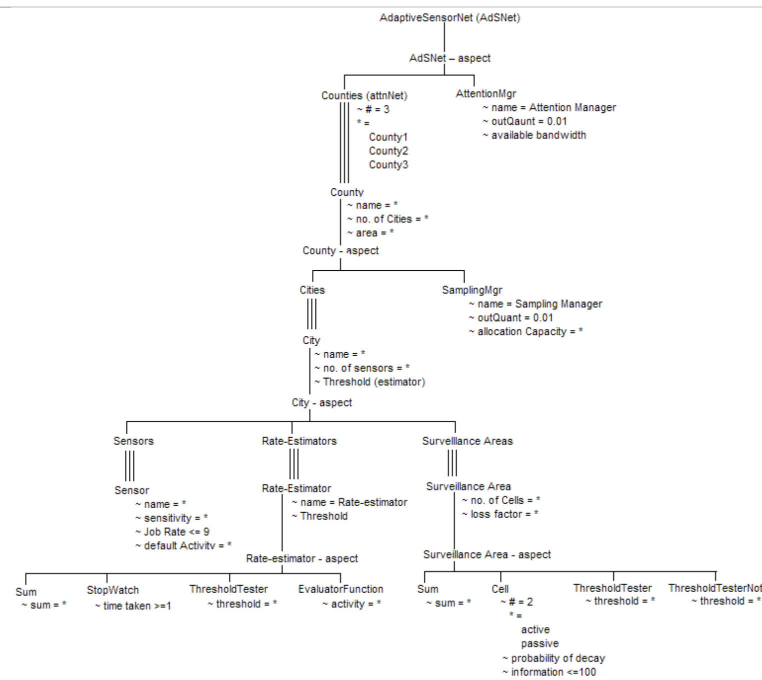

Figure 4. SES for *AdsNet* System

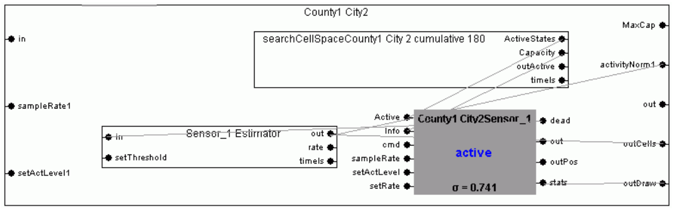

Figure 5. Sensor Estimator Pair

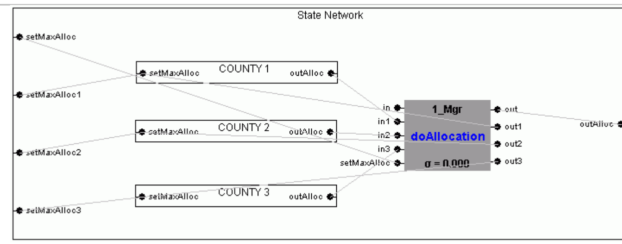

Figure 6. Top level view of a State constituting Counties

Various sensors are distributed in the area along with the local RAMs. This forms the periphery of the system. The demand for the resources travels upwards and the resources to address the "important" and "crucial" demands are addressed while maintaining a zero-sum game.

Table 2 lists various components in the system architecture and their role. For detailed DEVS behavior specifications, refer Mittal's thesis [10].

RE is a realization of context sensitivity guided by task-directed biasing. Every sensor is provided with a RE to *validate* the results of the sensor. Figure 5 shows the DEVS realization of City structure (realized from SES shown in Figure 4). For illustration, only one Sensor-RE pair is shown, which is coupled to a Surveillance Area comprising of 180 cells. Figure 6 shows three counties under the control of a single Attention Manager (also called Sampling Manager [RAM] at the County level), which assigns maximum allocation rate to counties. The manager may be considered a communication backbone as well that distributes bandwidth among different counties proportional to their total activity.

RAM is not responsible for how the resources are distributed inside the county and is totally ignorant of the structure inside the counties. The manager sees the counties as they appear at this level, a black-box. Each county shown above has a different behavior and is not identical to other counties in terms of cells. Each level has a different loss factor of how information gets transferred to the sensors.

Table 2. Component behavior specification

Entity	**Function**
COUPLED	
AdsNet	The topmost level in the sample system. Contains counties and an Attention manager, similar in functionality to the Sampling Manager
County	An intermediate level in a hierarchical system that contains cities and county-level Sampling Manager to allocate resources to various cities.
City	Comprises of at most two sensors (with accompanying Rate Estimator) attached to Surveillance Areas
Rate Estimator	"Thinking element" of a Sensor that determines the qualitative level of sensor data and reports it up the hierarchy
Surveillance Area	Composed of atomic Cell(s) and represents aggregate activity in a finite area. It behaves according to Qualitative System Specifications (QSS) and provides aggregated values not directly related to the individual threshold values of each individual Cell.
ATOMIC	
AttentionMgr	The behavior is identical to the Sampling Manager.
SamplingMgr	Control and allocates resources to be consumed by sensors in relation to their Surveillance Area. It assigns the sampling rate of sensors based on the information/activity reported to it. It also works on a threshold crossing concept. In an evolving system, it has the capability to alter the number of sensors.
Sensor	Monitors a Surveillance Area. It has its own sensitivity level and a maximum sampling rate. It receives the activity from the area and based on its sensitivity level, reports the accumulated activity to its Sampling Manager up the hierarchy and receives a new sampling rate.
EvaluatorFunction	Provides the evaluation criteria for accumulated activity of an area
Sum	Accumulates information (activity count) from all the active cells in an area. It is used at two levels of hierarchy, viz. Surveillance Area and Rate Estimator
StopWatch	Keeps track of time a particular area remains active (above threshold) and notifies the time only when the area becomes dead. It introduces loss factor in the amount of activity reported in the area.
ThresholdTester	Checks if the number of cells becoming active are over a certain threshold before it can trigger off the stop watch to notify that they are dead
ThresholdTesterNot	Keeps track of the situation when the threshold hasn't been reached and keeps the StopWatch in engaged state so that it continues to accumulate time
Cell	Turns active randomly and is proportional to the resources available in a Surveillance Area. It uses resources. It turn passive when resources turn zero.

Table 3. Model Design elements

Symbol	Name	Significance
A	Incoming activity	This is the count of all the active cells in a surveillance-area and abstracts the notion of group of neurons showing correlated firing.
S	Sampling rate of sensor	The rate at which a sensor samples activity. Higher rate implies high sensitivity and more resource usage
T	Rate-Estimator threshold	A numerical value that "validates" sensory output. This is an abstraction that implements salience or selective tuning
m	Number of data messages	Total number of messages flowing in the system at a given time
z	Number of house-keeping messages	Total number of messages that are required to keep the system active. This is analogous to various background processes that use resources for normal functioning of the active components
R	Job rate of sensor	The rate at which a sensor generates an output. This is directly proportional to the sampling rate
N	Number of message in network queue	Number of message in the network channels en-route various destinations. This is an abstraction of the conducting medium that amplifies/inhibits communication.
n	Number of sensors	Total number of sensors in the system

We expect a behavior of distribution of resources from this level to lower levels (inside the counties) till the allocation reaches the sensor that displays the highest activity. The algorithms implemented at different levels are independent of the hierarchy at every level as the RAM at that level distributes what's available with it. We observed a similar behavior when we ran experiments.

4. THEORETICAL ANALYSIS

4.1. Model Design

This section defines the boundary conditions for the model parameters involved. Table 3 lists parameters and their role. Detailed implementation can be seen in [10].

4.2. Presence of Rate-estimator (RE) with sensor

4.2.1. Estimator Threshold

The AdSNet System has a temporal character and has a working delay of 1 sec. This implies that any change in the activity has to persist for 1 sec to actually receive a new Sampling rate from the sampling manager (RAM). It is kept as such so that any small accidental change in the Activity measure doesn't produce change in the Sampling rate updates. This critical design consideration prevents the system from going into rapid oscillations. The system makes sure that the incoming activity has persisted long enough before it communicates this change to the next level. This buffering/working delay is based on the system requirements. The default sampling rate of a sensor is certainly less than 1 and consequently, the default Job-rate will be less than R. Therefore, the total amount of Activity reported per second is, $\xi = A \times R$.

RE reports this activity to the RAM when the activity surpasses the threshold T. As a result, if the delay factor is 1 sec, then, $T > \xi$. This condition will enable RE to report the activity after at least 1 sec (*iff* the sensor is operating at maximum sampling rate)

4.2.2. Number of messages in System (Bandwidth Usage)

The bandwidth usage is determined by the number of messages in the Network Queue at a particular instant of time.

N = No. of Data Messages + Housekeeping Messages

Consider the case when the RE is not present. In such a scenario, every data message (reporting of activity) from a sensor reaches RE.

Total data messages per sec, $m = n \times R$, where $R \leq \max Job\ rate$. Hence, Total messages in the network without the Estimator,

$$\boldsymbol{N_{\notin} = m + z} \tag{1}$$

Now, considering the case when RE is present. Using (1), total data message per sec = $n \times \frac{R}{T} = \frac{m}{T}$

All the data messages go from the sensor into the RE that produces only one data output once the threshold is reached. Consequently, the effective data message is reduced by a factor of T. Note that RE is an integral part of an intelligent sensor and the data message need not leave the boundaries of sensor to come in the network channel. An output from the RE is used for communicating information.

Total messages in Network Queue, $\boldsymbol{N_{\in} = \frac{m}{T} + z}$ (2)

Taking ratios of $N\in$ with $N\notin$, the ratio of Bandwidth usage for a system with RE and without RE is:

$$\gamma = \frac{N_{\in}}{N_{\notin}} = \frac{\frac{m}{T} + z}{m + z} = \frac{\frac{1}{T} + k}{1 + k}, \quad where\ k = \frac{z}{m}$$

Clearly, γ can be approximated to $\frac{1}{T}$ when $m >> z$ i.e. when the data messages are greater than the housekeeping messages. This is a practical assumption when the sensors are in a steady state and no housekeeping is required to maintain the system. Recall that in a discrete event scenario, message passing only occurs when there is a change in the state of the system. Consequently, $N_{\in} \cong \frac{1}{T} N_{\notin}$, which implies that the messages in the Network Queue are reduced by a factor of T, when RE is in operation. Another implication is: Having a "thinking component" that can put activity in "context" reduces the network chatter and conflicting information reaching higher levels of decision making.

4.3. Sampling algorithms

RAM at each hierarchical level implements the algorithm to assign the sampling rates to different sensors inside the area.

4.3.1. Normalized-Sum Rule (NS Rule)

The NS Rule is based on the condition of limited resources (the situation when there are no free resources available and all the resources are already distributed). RAM has fixed resources. Consequently, the most active sensor should be granted maximum amount of resources as compared to other sensors. Any increase in the sampling rate of any sensor brings about the decrease in the rates of other sensors and vice versa. The NS model is represented as:

$$S_k = \frac{A_k}{\sum_i^j A_i}. where\ i < k \leq j; and\ i, j\ \subset N\ and\ j \geq 2$$

4.3.2. Normalized-Max Rule (NM Rule)

The NM Rule is based on the condition of unlimited resources (situation when there is always an amount of free resources available with RAM). When the resources are freely available with RAM, it can distribute them to the sensors and let them sample at the maximum rate. This would enable each sensor working at its maximum sampling rate. In order to focus and bring attention to the highest activity or highly active sensor, all the activity reported by the sensors is normalized by the maximum activity amongst them. As a result, the sampling rate becomes a function of maximum activity present in the system at a given time. It can be represented as:

$$S_k = \frac{A_k}{A_{max}}$$

where A_k is the incoming activity and A_{max} is the current maximum activity. This rule has following properties:

I. The maximum activity at any given instant defines the state of the system. If any sensor has a high Activity, it forces other sensors in the neighborhood (under the same RAM) to increase their sampling rate (through RAM)
II. If $A_k < A_{max}$, the arrival of any new Activity does not disturb the configuration of the system as long as it is less than the current maximum value
III. If $A_k > A_{max}$, then the incoming Activity replaces the current value of $\boldsymbol{A}_{\max}$ and becomes $\boldsymbol{A}_{\max}$ itself, resulting in updation of all the sampling rates in the vicinity of this sensor. As a result, the new activity grabs hold of the maximum resources that it can utilize *iff* it has the maximum activity as compared to other sensors in the system.
IV. The assigned sampling rate is independent of the number of sensors operating at any given time in the system. As a result, the Job rate is also independent of the number of sensors. **The NM Rule promotes scalability**.
V. Each sensor can work to its maximum potential (producing the max possible Job rate) unlike the NS Rule (where it has an upper bound) irrespective of the system load.
VI. It helps us to define that **providing resources** and **providing attention** are two different operations. Providing attention leads to providing resources but not vice versa. Property (III) makes it more evident when the new incoming activity registers itself with RAM and causes reconfiguration of the entire sampling rate table based on its value. On the other side, property (II) provides the resources to the incoming activity but no attention.

4.3.3. Tunable Alpha-Beta Rule (TA Rule)

The TA Rule exploits the benefits of the NM Rule and adds negative feedback. It incorporates the previous sampling rate of the sensor (reporting the new activity) in the determination of the new sampling rate. Though the overhead increases at RAM's end as it has to maintain a separate table for storing the previous sampling rates, it makes the system more realistic. Keeping the previous sampling rate in the calculation brings an element of "inertia" in the system and incorporates history/memory as in various non-linear and natural systems. This inertia factor is tunable and is presented as follows:

Let the change in sampling rate be defined by $\frac{dS_k}{dt}$,

$$\frac{dS_k}{dt} = \frac{A_k}{A_{maz}} - S_k$$

$$S_k\ (t + dt) = S_k + dt \left(\frac{A_k}{A_{max}} - S_k \right)$$

$$= S_k(t)(1 - dt) + dt \left(\frac{A_k}{A_{max}} \right)$$

which gives us a new sampling rate of sensor *k*,

$$S_k' = \alpha S_k + \beta \frac{A_k}{A_{max}}$$

The TA Rule has the following properties:

I. Based on the values of α and β, the system can be made more sensitive or more sluggish. α and β are correlated by the function $\alpha+\beta = 1$

II. Increasing the sensitivity (β) automatically reduces the inertia factor (α)

III. This rule makes the transition to a new activity level smoothly. There are no sharp rises and falls in the values of new sampling rate when compared with previous sampling rate.

IV. Higher the β, the faster the responsiveness of the system (through RAM) and higher the α, the more inertia the system has and more averse to change to a new sampling rate.

V. Striking a balance between α and β defines the responsiveness of the system.

5. QUANTITATIVE RESULTS

This section provides simulation results for NS, NM and TA algorithms within the RAM at the topmost RAM (Fig. 3).

5.1. Response Time v/s new incoming activity

The environment consists of five sensors at their default activity value 100 and RE threshold $T = 3500$. Maximum job production rate R = 9 jobs/sec. The experiment is done for both NM Rule and NS Rule. The analysis yields a chart between Response Time of AdsNet System against a new incoming activity (Figure 7). Response Time is defined as the time taken by the system (specifically, RAM) to identify the most-active sensor and direct resources to the corresponding sensor, thereby increasing the sampling rate of that sensor.

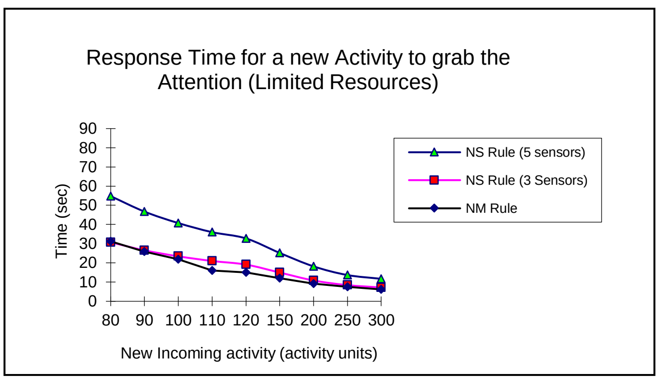

Figure 7. Response Time v/s Incoming Activity

Comments and Inferences:

a) NS Rule has a higher Response Time as the number of sensors increases while NM Rule is independent of number of sensors in the network

b) The AdSNet system works well with a network model and an activity can break-into the system after passing through the network channel that induces losses as the information travels up the hierarchy and/or through a conducting medium.

5.2. Response Time with TA Rule

This experiment evaluates TA Rule. The experiment setup is as follows: All the rate-estimators have threshold T of 3500. The maximum job rate R is 9 jobs/sec and all the other parameters have their usual meaning. The analysis of this rule yields two charts. The first chart (Figure 8) is of Response Time v/s the new incoming activity.

Comments and Inferences:

a) When the incoming activity value is same as that of the background sensor activity (value 100), the Response time takes maximum value (when compared to Response time values for activity higher than background activity), irrespective of α value.

b) α= 0.9 takes the least Response time and there is a marked difference in Response time values for $\alpha > 0.5$.

c) Response time increases for $0.1 \le \alpha \le 0.7$ and for $\alpha > 0.7$, Response time starts decreasing.

The second chart (Figure 9) yields Stabilization time, defined as the time taken by the RAM to overcome the inertia as the sampling rate is increased gradually for the corresponding sensor.

Comments and Inferences:

a) Stabilization Time again takes a steep curve at value equal to 100 indicating the competing nature of the new incoming activity and is same for every value of α when the incoming activity is same as that of background activity.

b) Stabilization time decreases when α increases from 0.1 to 0.7 ($0.1 \le \alpha \le 0.7$) and starts increasing for $\alpha \ge 0.7$.

c) Stabilization time for α= 0.9 is unusually high

d) $\alpha = 0.7$ has the lowest Stabilization time.

e) Stabilization time for $\alpha = 0.8$ is in the same range as the Stabilization time for $0.1 \le \alpha \le 0.7$

Figure 8 and Figure 9 bring about two surprising results:

a) Response time is lowest when α is at maximum, which is contrary to the TA Rule that says the higher the α, more sluggish the system becomes.

b) The system is not at its maximum sensitivity (or lowest Stabilization time) when α is lowest i.e. 0.1 (β is highest simultaneously as $\alpha + \beta = 1$)

However, both the above anomalies are completely in accordance with the definition of TA Rule. If we consider the sum of Response Time and the Stabilization Time, higher α does end up making the system sluggish and lowest α (highest β) does makes the system more responsive and bringing it quickly to steady state. In order to tune the values of α and β such that we can have minimum Response time as well as optimum Stabilization time, we have to exclude all $\alpha > 0.7$.

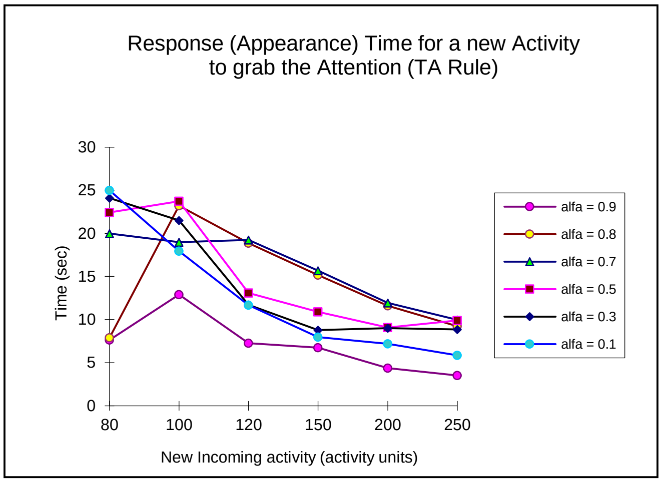

Figure 8. Response Time for Incoming Activity with respect to different Alfa values

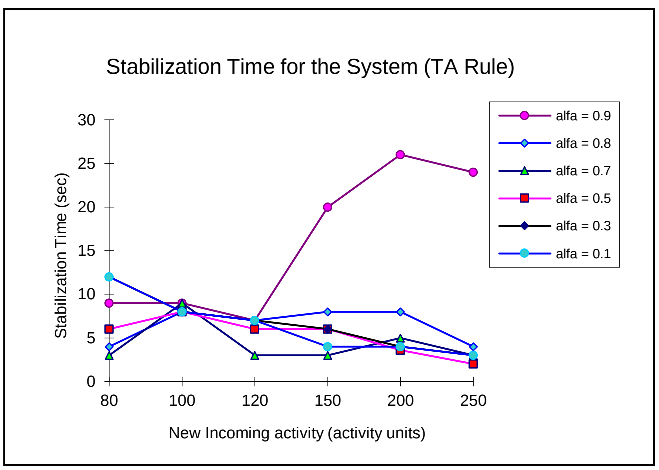

Figure 9. Stabilization Time for incoming activity with respect to different Alfa values

The simulation results indicate that the range of α must confirm to the condition $0.1 \leq \alpha \leq 0.7$ for both Response time and Stabilization time behavior to coexist and comply with the requirements of an attention management system. This result is currently empirical and further analysis is needed to establish the significance of $\alpha \leq 0.7$. The stochastic nature of the activity areas also needs to be further investigated and will be reported in our extended article.

6. CONCLUSIONS AND FUTURE WORK

This research has successfully implemented a modular system capable of focusing attention to components displaying high activity and directing resources towards them so that they can accomplish their task efficiently. The framework has been built using DEVS formalism. Exploitation of variable structure DEVS allows the system to exhibit both the adaptive behavior as their environment changes and the steady state in a dynamic environment [11]. The architecture design can be mapped to any real life system that is hierarchically organized working along the rule of “chains of command”, which implies that the information is filtered and condensed as it travels up in the hierarchy. To demonstrate the basic concepts, the system has been mapped to geographical area distribution under the control of managers that allocates resources (like bandwidth and channel capacity) to areas under their control, which in turn distribute the resources to the end-user (sensors). In other examples, resources may take the form of knowledge partitioning as applicable in problems dealing with Big Data and Genetic Algorithms. The distribution is done intelligently with the most active component receiving the maximum number of resources. As the components pass through cycles of high and low activity so does the assignment of the resources allocated to it. The simulation results have confirmed that the system is capable of directing and switching its focus to components that display persistent high activity during simulation and also can withdraw attention from components which are not displaying any activity. Reference [13] shows that the reduction in energy in actual implemented systems (e.g., hardware) can be measured and compared to the ideal level given by the disparity measure discussed above. Likewise, the attention switching architectures saves energy by not wasting it on components that do not need it in dynamic fashion, the (non-functional) performance of this architectures can be gauged by the disparity measure, a task left open for continued research.

This work adapts the WTA algorithm by incorporating various sampling algorithms that determine where an activity of highest importance, thereby receiving attention from DDM or a RAM. The criteria for deciding an activity “important” is based on the sensitivity of the sensor and the RE threshold. This component validates the importance of any activity sensed by the sensor. The WTA model enables the system to continually shift focus and direct its attention to the most active component. This model is an abstracted version of RCIDS, where the resources are directed towards a focused activity. This makes the system a generalized architecture capable of focusing attention and concentrating on the job at hand by providing more resource to it by redistribution and reallocation. It calls for two entities in the system:

1. Rate Estimator: Capable of drawing conclusions and analyzing situations based on threshold mechanisms in sensor's local environment
2. Sampling/Resource-allocation Manager: Capable of focusing attention to the most "relevant" sensor by increasing the sampling rate of the corresponding sensor.

The system has a fractal nature where there exists a RAM at every level to direct focus and attention and an RE coupled to every sensory element. The RE may or may not be present at the intermediate levels in the hierarchy but it must be at the coarsest level, to deduce and validate what the sensors are witnessing. The system also allows resources and peripheral attention to the ongoing working sensors and doesn't inhibit or stall their operation in the pursuit of focusing attention to the important one. For different WTA mechanisms, the sensor population is met accordingly and in no case, the resources are completely withdrawn from the running sensors as it is not predictable which sensor might produce an important information the next instant. The system lets the other sensors keep working at their default settings and provide the resources for their operation and intermittently switches when an activity of high importance is encountered and advertised by any sensor.

We discussed the nature of attention and develop a system model that validates its emergent nature. This research also validates that in a resource constrained environment the system has to switch attention to focus the task-at-hand and to live within the energy constraints it has. We developed two metrics, i.e. Response time and Stabilization time that quantifies the time taken for the system to switch attention and continue to pay attention.

The component-based architecture and distributed operation of the system facilitates its deployment in the real-world in terms of federates and participate in larger system of systems [15]. The AdsNet system is a proof-of-concept model for the larger RCIDS in the fact that the designed model is built using DEVS formalism suitable for modeling complex dynamical systems. The system can be made predictive and robust with more detailed modeling of the RE and WTA mechanisms implemented in the RAM, supported with efficient synchronization strategies.

7. REFERENCES

[1] Barabási, A.L. "Linked: How everything is connected to everything else and what it means for business, science and everyday life". New York, NY, Penguin Books. 2003

[2]. Tsotsos, J.K., Bruce, N.B., "Computational Foundations for Attentive Processes", Scholarpedia, 3(12):6545, doi:10.4249/scholarpedia.6545, 2008

[3]. Tsotsos, J.K., Rothenstein, A., "Computational models of Visual Attention", Scholarpedia, 6(1):6201, doi:10.4249/scholarpedia.6201

[4] Tsotsos, J.K., "Computational Resources do constrain behavior", Behavioral Brain Sc., pg 506-507, 1991

[5] Tsotsos, J.K., "Modeling Visual Attention via Selective Tuning", Artificial Intelligence 78, 507-545, 1995

[6] Tsotsos, J.K., "Limited Capacity is a Sufficient Reason for Attentive Behavior", Cognition and Consciousness 6, (invited Commentary on A.H.C. van der Heijden and S. Bem), p.429 - 436, 1997

[7] Styles, E., "The Psychology of Attention", Pyschology Press, 1997

[8] Shipp, S., "The brain circuitry of Attention", Trends in Cognitive Sciences 8(5), p223-230., 2004

[9] Pinker, S, How the Mind Works, W.W. Norton, New York, NY, 1997

[10] Mittal, S., "Attention-Focusing Architecture for Scalable Networked Systems using Discrete Events Systems Formalism", MS Thesis, University of Arizona, Tucson, 2003.

[11] Mittal, S., "Emergence in Stigmergic and Complex Adaptive Systems: A Discrete Event Systems perspective", Special Issue on Stigmergy, Cognitive Systems Research, 2012.

[12] Hu, X, Zeigler, B.P. "Linking Information and Energy – Activity-based Energy-aware Information Processing", SIMULATION, 2012

[13] Zeigler, B.P. "Activity-based Implementation of Systems of Systems", in *Guide to Modeling and Simulation of System of Systems (Simulation Foundations, Methods and Applications)* by Bernard P. Zeigler, Hessam S. Sarjoughian, Raphel Duboz, Jean-Christophe Soulie, Springer, 2012

[14] Zeigler, B.P., Praehofer, H., Kim, T.G., "Theory of Modeling and Simulation", Academic Press, New York, NY, 2000

[15] Mittal, S., Martin, J.L.R. "Netcentric System of Systems Engineering with DEVS Unified Process", Boca Raton, FL., 2013

Biography

SAURABH MITTAL is the founder and principal of Dunip Technologies. He received both PhD (2007) and MS (2003) in Electrical and Computer Engineering from the University of Arizona, Tucson. He is also affiliated with L3 Communications, Link Simulation & Training. He can be reached at smittal@duniptech.com.

BERNARD P. ZEIGLER is the Chief Scientist at RTSync Corp. His bio is available at: http://en.wikipedia.org/wiki/Bernard_P._Zeigler. He can be reached at: zeigler@rtsync.com.