Publication of Little Lion Scientific R&D, Islamabad PAKISTAN
Journal of Theoretical and Applied Information Technology
15th March 2011. Vol. 25 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

WIRELESS SENSOR NETWORK SIMULATION OF THE ENERGY CONSUMPTION BY A MULTI AGENTS SYSTEM

[1]Rahal ROMADI, Hassan Berbia, B.Bounabat
L2MI, Ecole Nationale Supérieure d'Informatique et d'Analyse des Systèmes
BP 713, Agdal Rabat, Maroc. Tél: (212) 7 77 85 79 Fax: (212) 7 77 72 30
E-mail: romadi@ensias.ma, berbiaa@ensias.ma, bounabat@ensias.ma

ABSTRACT

A Wireless Sensor Network (WSN) is one that is in continual interaction with its environment, and executes at a pace determined by that environment. The use of rigorous formal method in specification and validation can help designers to limit the introduction of potentially faulty components during the construction of the system.

Due to their complex nature, WSN are extremely difficult to specify and validate. In this paper, we propose a new formal model for the specification and the validation of such systems. This approach considers a WSN as a Reactive Multi-Agent System consisting of concurrent reactive agents that cooperate with each other to achieve the desired functionality. In addition, this approach uses formal synchronous specification and verification tools in order to specify and to verify the systems behaviors.

Keywords: Wireless Sensor Network, Reactive Systems, Reactive Agent, Specification, Formal Methods, Verification.

1. INTRODUCTION

A reactive system is one that is in continual interaction with its environment, and executes at a pace determined by that environment. Thus, reactive systems are complex computer systems, and may not be modeled by transformational techniques. The use of rigorous formal methods in specification and validation can help designers to limit the introduction of potentially faulty components during the construction of the system. Specification modeling is an important stage in reactive system design where the designers specify the desired properties in the form of a specification model that acts as the guidance and source for the implementation.

Validation of an abstract specification of a reactive system, is an important aspect of system design. One approach for validation is to consider observable behavior as criteria to determine success.

In this paper, we propose a formal model for the specification and the validation of reactive system. This approach considers a reactive system as a Reactive Multi-Agent System, i.e a distributed computing system consisting of several autonomous reactive agents (as computing units) that coordinate their action in order to fulfill usually joint but also sometimes competitive tasks. Concurrency is further characterized by the need to express communication and synchronization among concurrent agents.

In recent years, a lot of research has been conducted on an important class of reactive system, i.e., wireless sensors, which will constitute the infrastructure for the ambient intelligence vision.

This new kind of embedded systems has great potential for many applications, for example surveillance, disaster relief applications, environment monitoring, emergency medical response and home automation etc.

Given its numerous domains of application, industry is also beginning to express its interest in wireless sensors. However, the great mismatch between research at different levels (application, network, node) has forced industry to be reluctant to use research results. This study is part of greater initiative to narrow this mismatch. Power management in WSNs is a very rich area, since the matter is of utmost importance in this field. Motivations for the extensive research carried out in this field are numerous. Firstly, the battery is generally not replaceable, due to the randomness of the sensing device's position and sometimes also to the dangerousness of the sensing field.

Publication of Little Lion Scientific R&D, Islamabad PAKISTAN
Journal of Theoretical and Applied Information Technology
15th March 2011. Vol. 25 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Therefore, battery lifetime is synonym of sensor lifetime and must be extended as much as possible. Secondly, the progress made in battery capacity, lifetime and size is at best limited as compared to the one in processing power, storage capacities and size. Lastly, with performance constraints ever-increasing, power management is guaranteed to always need improvement.

L'objectif est de calculer la durée de vie du WSN par une simulation de son comportement.

2. SPECIFICATION AND SIMULATION TOOLS

This section will describe all the specification and verification tools used in this work.

2.1 Syncharts

SYNCHARTS (SC) are introduced by Harel [2][3] like a visual formalism that provides a way to represent state diagrams with notions like hierarchy, concurency, broadcast communication and temporized state. A SC can be seen like one or several automata which are labeled by ?event[condition]/!action. SC is said to be synchronous because the system reacts to events by instantly updating its internal state and producing actions, the actions produced can trigger in the same instant other transitions, this is named chain reaction causing a set of transitions, the system is always in a waiting state until the condition for a transition is true.
In the method presented here, the syncharts are used to specify reactive behaviors. The hierarchical aspect allows us to represent several levels of abstraction. To validate and simulate these specifications, they are automatically translated into Esterel.

2.2 Esterel

ESTEREL [4][5][6] is a language, with precisely defined mathematical semantics, for programming the class of input-driven deterministic systems. The software environment of ESTEREL provides high-quality tools, including an editor, compiler, simulator (XES tool), debugger and verifier.

2.3 Real-time temporal logic

Temporal logic has been widely used for the specification and verification of concurrent systems. However, these temporal logics only allow qualitative reasoning about time. Several extensions have been proposed for expressing and reasoning about real-time systems. These include Real-Time Temporal Logic (RTTL), which is based on linear time temporal logic, and allows in addition the expression of quantitative real-time properties (e.g. exact delays or event deadlines).

Example of RTTL Formula

$s_1 \wedge t = T \rightarrow \Diamond (s_2 \wedge t \leq T + 5)$ - If s_1 is true now and the clock reads T ticks, then within T + 5 clock ticks, s_2 must become true. Thus, once s_1 becomes true, s_2 must become true no more than 5 ticks later. This formula can be also written as follows: $s_1 \rightarrow \Diamond_{[0,5]} s_2$ or $s_1 \rightarrow \Diamond_{<=5} s_2$

The formula $s_1 \leftrightarrow s_3$ indicates that events s_1, s_3 are simultaneous. If C(w) is a RTTL formula defining a temporal constraint on an event w, then w $\models$ C(w) means that w satisfies the formula C(w).

3. REACTIVE DECISIONAL AGENT

In this paper, the agents are classed as either deliberative or reactive [9][10]. Deliberative agents derive from the deliberative thinking paradigm : the agents possess an internal symbolic, reasoning model and they engage in planning and negotiation in order to achieve coordination with other agents. Reactive agents don't have any internal symbolic models of their environment, and they act using a stimulus/response type of behavior by responding to the present state of the environment in which they are embedded.
The proposed model of reactive agent consists in putting forward decisional models allowing the representation of objects according to their behavioral aspects and their degree of intelligence.
Definitions. A Reactive Decisional Agent (RDA)[9][10] is 9-tuple noted < Id, A, D, S, E', O, O', act, dec, sig > where :

- Id : agent identity
- A : Set of actions exerted on the agent. Each action, undergone by an object, represents a possible operation to be carried out on this object in order to achieve a specific goal.
- D : Set of decisions generated by the agent. Each decision is a solution concerning process behavior in the future; each

Publication of Little Lion Scientific R&D, Islamabad PAKISTAN
Journal of Theoretical and Applied Information Technology
15th March 2011. Vol. 25 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

decision is characterized by its action horizon : Ha, the time during which this decision remains valid.

- S : Set of Signaling received by the agent. Each Signaling received by an object, reflects at any given time the state of the controlled tools used to achieve a specific goal.
- E': Set of external states delivered by the agent. Each one represents the object state emitted to the environment.
- E : Set of agent's internal states. Each one indicates the current state of the agent.
- O: Set of agent's internal objectives. Each decision is elaborated in order to achieve an internal objective according to the current external objective and the actual internal state.
- O': Set of agent's external objectives which can be achieved. These objectives represent the agent's interpreting of each action.

From a dynamic point of view, the sets above indicate the received events (A, S), the emitted events (D, E') and the internal events (E,O,O').

Decisional Functions. act, dec, and sig are three decisional functions that define the behavior of a RDA.

act : $A \longrightarrow O'$

$a \longrightarrow o'$ with,

$\forall a \in A, \exists! o' \in O' / o' = act(a) \Rightarrow a \leftrightarrow o'$ (1)

(1) means that the occurrence of an action *a* implies instantaneously the occurrence of its associated external objective *o'* by the function *act*.

dec : $O' \times E \longrightarrow D \times O$

$(o', e) \longrightarrow (d, o)$ with,

$dec(o', e) = (d, o) \Rightarrow [o' \wedge e \leftrightarrow d \wedge o]$ (2)

(2) means that depending of the current external objective *o'* and as soon as the agent is in an appropriate internal state *e*, corresponding decision *d* an internal objective *o*, by the function *dec*, are instantaneously produced.

sig : $O' \times O \times S \longrightarrow E \times E'$

$(o', o, s) \longrightarrow (e, e')$ with,

$sig(o',o,s)= (e, e') \Rightarrow [o' \wedge o \wedge s \leftrightarrow e \wedge e']$ (3)

(3) means that depending of the current external objective *o'* and the expected internal objective *o*, and as soon as the receipt of a signaling *s*, its associated external state *e'* is instantaneously emitted and the new agent internal state becomes *e*. e' peut etre égal à 0, dans le cas où le signal ne necessite pas un etat externe.

Il y a trios état externs particuliers :

- Id_Recv_a : acquitte la reception de l'action a par l'agent d'identité Id
- Id_Ack_a : signal la réalisation avec succés de l'action a par l'agent Id
- Id_Timeout_a : signal que la réalisation de l'action par l'agent Id a echoué

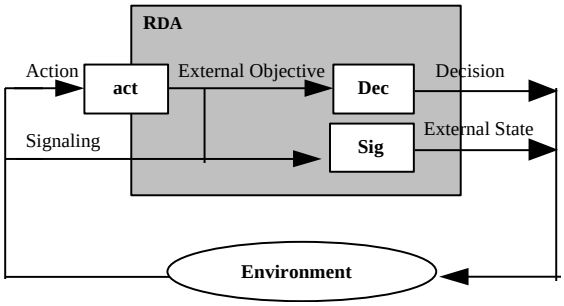

Fig.1. According to the formal definitions above, figure.1. shows the internal structure of a Reactive Decisional Agent. Act interprets an action as an external objective, that it used by Dec an Sig to generate agent appropriate responses.

Internal Architecture of an RDA. This section presents a set of SC which describe the external objective of a RDA.

External Objectives Manager. A Reactive Decisional Agent has an External Objective Manager. It consists in a SC model of the function *act* described above (Fig. 2). Each state represents an external objective whose activation is started by the reception of a specific action (*?Action*), and terminated by the emission of the acknowledgment external state (*!ExternalObjective*).

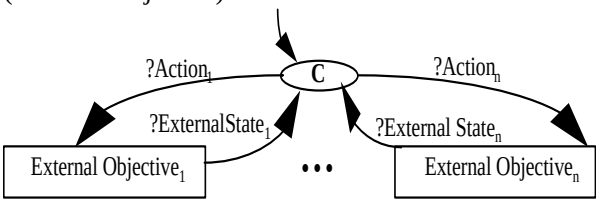

Fig. 2 . This shows a figure consisting of a SC model of External objectives manager.

In addition, each operating mode of the agent (normal mode, diagnostics modes, etc.) can be

Publication of Little Lion Scientific R&D, Islamabad PAKISTAN
Journal of Theoretical and Applied Information Technology
15th March 2011. Vol. 25 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

considered as an external objective to be reached. The objectives manager has to maintain the same objective or to change it, according to the occurred fault or failure.

*External Objectives Modeling***.** An external objective is composed by many others SC states corresponding to the associated internal states and internal objectives that are deducted by the functions dec and sig definitions (Fig. 3).

The transition (*Internal state* → *Internal objective*) is made by a decision emission (*!Decision*), and the transition (*internal objective* → *Internal state*) is made by a signaling receipt (*?S_OK*), and eventually an external state emission (*!e'*). Internal state C corresponds to the default initial state of a SC model. Internal state and Internal objective are indicated respectively by e_i et o_i. In case of an action horizon exceeding without receiving any acknowledgment signaling, the agent's internal state changes from e_i to eb_i (*breakdown state*).

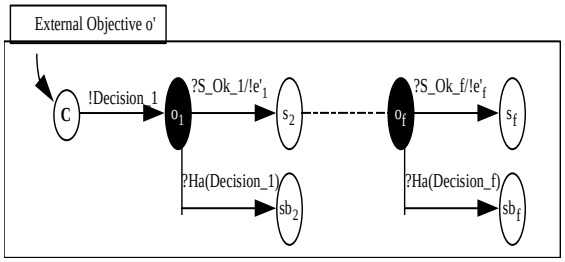

Fig.3. This figure shows the general SC model of an External objective.

3.3 Temporal constraints of an RDA

Decision Temporal Constraints. Each decision is characterized by its action horizon, Ha : the time during which this decision remains valid. So, an occurrence of a decision requires the occurrence of its corresponding acknowledgment signaling, in a delay that doesn't exceed its action horizon.

This defines the following function, acqDec :

$$acqDec : D \longrightarrow S \times IN$$
$$d \longrightarrow (s, Ha) = acqDec(d), \text{ with}$$
$$acqDec(d) = (s, Ha) \Rightarrow [d \rightarrow \Diamond_{<=Ha} s \,] \qquad (4)$$

In the following sections and for any decision d :

- acqDec(d) indicates the acknowledgment signaling of d,
- Ha(d) is the action horizon of d,
- C(d) points out the constraint $[d \rightarrow \Diamond_{<=Ha(d)} acqDec(d)]$

The temporal property that a RDA must verify :

$$\forall d \in D, d \Vdash C(d) \qquad (5)$$

$$timeout: O' \longrightarrow IN$$
$$card(D_{o'})$$
$$o' \longrightarrow \sum_{i=1} Ha(d_i), \text{ where } d_i \in D(o')$$

i.e. after an occurrence of an external objective o', the agent must generate the corresponding acknowledgment, in a delay that does not exceed timeout(o').

4. SPECIFICATION OF WSN

The internal organization of a WSN [11] consists in a tree, that is made up in parallel of a supervisor (*Supervisory Agent*), of two or several sub-agents components, and 3 communication interfaces between the supervisor and the sub-agents (fig 4).

Such system interacts with its environment by the means of :

- Actions exerted by this environment.
- External States emitted to the environment.

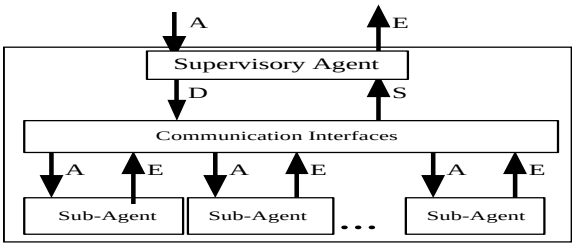

Fig.4. Communication Interface (CI)

4.1 Supervisory and Sub-Agents Levels. The supervisory agent (SRDA : Supervisory Reactive Decisional Agent) is a RDA controlling the component sub-agents, in order to achieve a goal or to solve a given problem.

This agent will manage the sequences of activation and the definition of the controlled sub-agents objectives. This management depends on :

- the actions exerted by the environment,
- the events generated by the sub-agents activities,
- the temporal constraints specific to any reactive system.

In addition, a WSN can be summarized with a simple SRDA directly connected to the controlled process. Each sub-agent can be considered as a reactive system. Thus, its internal structure is composed by its own SRDA, communication

Publication of Little Lion Scientific R&D, Islamabad PAKISTAN
Journal of Theoretical and Applied Information Technology
15th March 2011. Vol. 25 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

interfaces and sub-agents. A sub-agent objectives are to carry out sequences of tasks in response to any temporal constrained action exerted on him by the higher level.

4.2 Agent Interfaces.

The agent interfaces are of 3 types: decisional interface (Top/Down), signaling interfaces (Bottom/Up) and communicating interfaces.

- Decisional interface (DI) that translates a decision (d) generated by the SRDA into several actions (a_i), each one of them is intended for a sub-agent of the lower level.
- Signaling interface (SI) that synchronizes the external states (e'_i), sent by each sub-agent, and emits one signaling (s) intended for the SRDA.
- Communicating interface (CI) : the different agents communicate by diffusion. The communication interface specifies the channel access protocols (MAC) and routing protocol used. Initially, we will use an Esterel module that uses a random function to determine the channel state: before sending data agent must ensure that the channel is free if not it must waits a random time before trying again.
  For routing, we consider that all the information collected should be sent to a central agent (sink). Data of the agents that are not within the scope of the sink will be forwarded step by step by the other agents. The communication interface also allows an agent to specify the agents that are within the scope of his radio (fig 4).

4.3 Temporal properties

Through the notion of an action horizon (Ha) of a decision, the time during which the decision remains valid, the RDA-based specification of a WSN system ensures that the elements will have time periods coherent with the decision made by the agent, and coherent with the time periods of decisions made at lower levels of the hierarchy. The higher an agent is in the hierarchy, the greater the action horizon (Fig. 5).

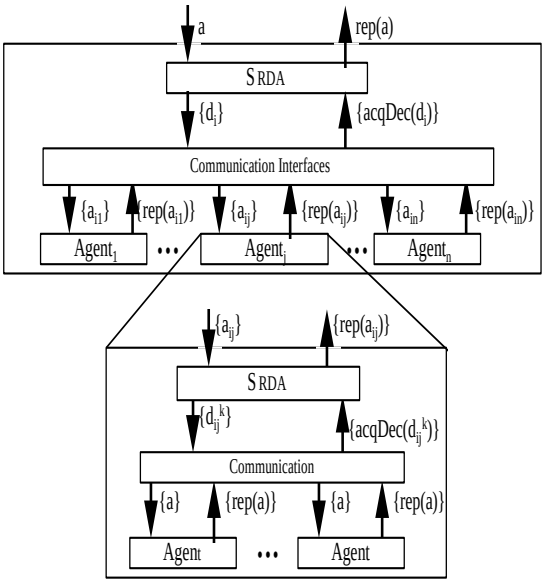

Fig.5. Flow of information inside a SMA formed by ARDC agents, inside a SMA formed by ARDC agents, the top-down flow consists in actions (a, a_{ij}) and their associated decisions (d_i, d_{ij}^k). The bottom-up flow consists in external states ($rep(a)$, $rep(a_{ij})$) and their associated signaling ($acqDec(d_i)$, $acqDec(d_{ij}^k)$).

The temporal constraints must be checked on each hierarchical level. The recursive character of this structure makes it possible to generalize the results obtained for only one hierarchical level. Thus, we can prove by deduction and according to notations of fig. 5:

$$d_{ij}^k \models C(d_{ij}^k) \qquad a \models C(a)$$

5. APPLICATION

We use a representative application of long lived, unattended wireless sensors. it is a wsn composed by four nodes and the management node (sink). Each node uses six peripherals: two SPI devices (Radio/Flash), two I2C sensors(Humidity/Temp) and two ADC sensors (Total solar/Photo Active). The two I2C sensors are on the same chip (same I2C address), but require separate sensing command sequences.

The measurements in Table 1[12] provide the basic means for calculating the energy costs of different I/O operations and sleep states.

The processor values are reported for each power state and include leakage currents of platform peripherals such as sensors, USB, and flash. The peripheral values do not include the processor

Publication of Little Lion Scientific R&D, Islamabad PAKISTAN
Journal of Theoretical and Applied Information Technology
15th March 2011. Vol. 25 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

current draw; instead, they show the lowest power state the processor can enter while that peripheral is in use. This is LPM3 for the radio because the SPI bus is off except for a few hundred microseconds of radio commands. It is LPM1 for the flash because logging operations keep the SPI bus on. It is also LPM1 for the analog sensors because the ADC requires a clock source, while it is LPM3 for the two I2C sensors because the bus is in software on GPIO pins. Finally, the voltage reference requires a 17ms warmup time before it can be used.

Device	Current	Time	Power state
Microcontroller			
Active	1,92mA	NA	NA
LPM1	182 uA	NA	NA
LPM3	9 uA	NA	NA
Vref On	536 uA	NA	NA
Radio			
Receive (LPL check)	18,86 mA	5ms	LPM3
Send (1 packet)	18,92 mA	12ms-1s (LPL)	LPM3
Flash			
Read Record	1,75 mA	5ms	LPM1
Write Record	2,69 mA	5ms	LPM1
Analog sensors	1,46 mA	2ms	LPM1
Humidity sensor	458 uA	75ms	LPM3
Temperature sensor	458 uA	220ms	LPM3

Table 1: Current draw, duration, and the lowest MCU power state of the major components used in our example application

Model Multi-agent (fig 6)

Every five minutes, the application samples four sensors and logs the readings in flash. Every twelve hours, the application retrieves new readings from flash and sends them to a gateway. The application logs values to flash to provide data reliability in case of temporary or long-term disconnection, a common problem in long-term deployments [12]. Sampling and sending are completely decoupled: the two parts have a producer/consumer relationship on the log.

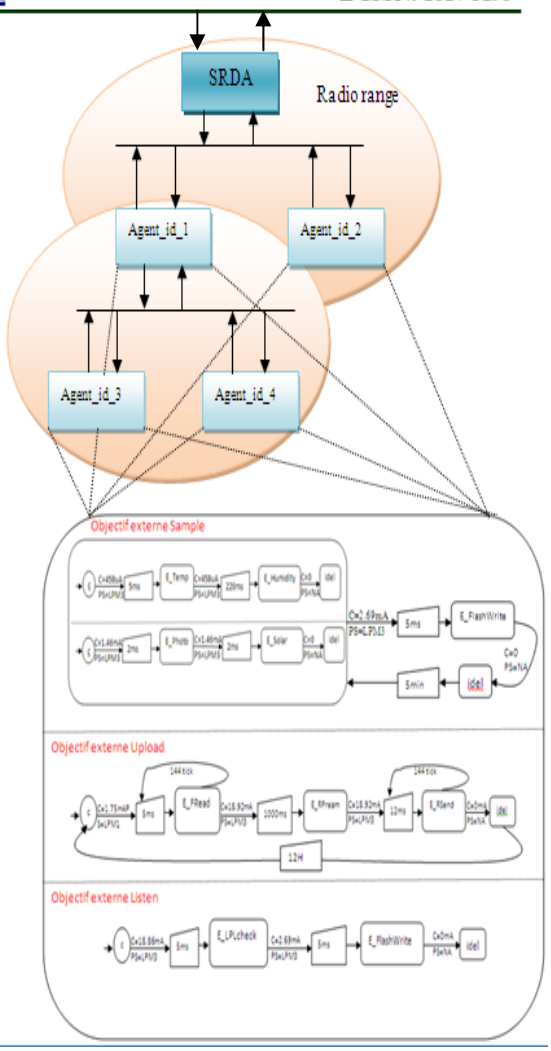

Fig 6: the WSN nodes have the same behavior that was specified by a syncharts (sc): state and because the three external objectives evolve on Parallel

Simulation of the model

The approach adopted here is to translate the specified SC behaviors to the synchronous language ESTEREL. According to automated translation tool developed in [17], the mapping of a modeled reactive system in ESTEREL is done easily by translating the communication interfaces (ID, IS), the supervisory agent (sink) and the sub-agents. The ESTEREL code associated to model:

Module wsn :

Run agent [signal Sample/Sample_id1, Sample_Temp/Sample_Temp_id1, Sample_Humidity/ Sample_Humidity_id1, Sample_Photo/ Sample_Photo_id1,Sample_TotalSolar Sample_TotalSolar_id1, Flash_Write/

Publication of Little Lion Scientific R&D, Islamabad PAKISTAN
Journal of Theoretical and Applied Information Technology
15th March 2011. Vol. 25 No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

```
Flash_Write_id1, Flash_Read Flash_Read_id1,
Radio_Send/ Radio_Send_id1 ]
||
Run agent [ signal Sample/Sample_id2,
Sample_Temp/Sample_Temp_id2,
Sample_Humidity/ Sample_Humidity_id2,
Sample_Photo/ Sample_Photo_id2,
Sample_TotalSolar /Sample_TotalSolar_id2,
Flash_Write/ Flash_Write_id2, Flash_Read/
Flash_Read_id2, Radio_Send/ Radio_Send_id2 ]
||
Run agent [ signal Sample/Sample_id3,
Sample_Temp/Sample_Temp_id3,
Sample_Humidity/ Sample_Humidity_id3,
Sample_Photo/
Sample_Photo_id3,Sample_TotalSolar/
Sample_TotalSolar_id3, Flash_Write/
Flash_Write_id3, Flash_Read/ Flash_Read_id3,
Radio_Send/ Radio_Send_id3 ]
||
Run agent [ signal Sample/Sample_id4,
Sample_Temp/Sample_Temp_id4,
Sample_Humidity/ Sample_Humidity_id4,
Sample_Photo/
Sample_Photo_id4,Sample_TotalSolar/
Sample_TotalSolar_id4, Flash_Write/
Flash_Write_id4, Flash_Read/ Flash_Read_id4,
Radio_Send/ Radio_Send_id4 ]
end
module agent :
input seconde, minute,hours;
var Nbr_Sample :=0 : integer in
output Sample, Sample_Temp,
Sample_Humidity,
Sample_Photo,Sample_TotalSolar, Flash_Write,
Flash_Read, Radio_Send,
signal ESample :=0 ,EFlash_Read
:=0,EFlash_Write :=0, ERadio_Send:=0 : integer
in
every 5 minutes do
  Nbr_Sample++;
  abort
        sustain sample;
   when idel
||
   [
        emit Sample_Temp;
        emit ESample(?pre(ESample)+100,76);
        emit Sample_Humidity;
        emit ESample (?pre(ESample)+34,35);
      ]
    ||
      [
        emit Sample_Photo;
        emit ESample (?pre(ESample)+2,92);
        emit Sample_TotalSolar;
        emit ESample (?pre(ESample)+2,92);
      ];
      emit Flash_Write;
emit EFlash_Write(?pre(EFlash_Write)+13,45);
      emit idel;
end every
||
[ every 12 hours do
abort
   sustain upload;
when end_send
||
loop
     emit Flash_Read;
 emit EFlash_Read (?pre(EFlash_Read)+8,75);
     emit Radio_Send;
     emit
ERadio_Send(?pre(ERadio_Send)+0,23);
     Nbr_Sample--;
when (Nbr_Sample !=0);
emit end_send;
end every
]
end signal;
end var;
end modul.
```

Result of simulation:

Per-day energy consumption of each agent in µAs

Data Logging:	**(µAs)**
$E_{FlashWrite}$	13.45
E_{Temp}	100.76
$E_{Humidity}$	34.35
E_{Photo}	2.92
$E_{TotalSolar}$	2.92
E_{LPM3}	2.574
E_{LPM1}	1.638
Total per sample	**168**

Data Upload:	**(µAs)**
$E_{FlashRead}$	1260
$E_{RadioSend}$	32, 694
E_{LPM3}	18
E_{LPM1}	131
Total per send	**53, 023**

Listening:	**(µAs)**
$E_{LPLcheck}$	94.3
E_{LPM3}	0.045
Total per check	**94.3**

Total Day	**agt_id1**	**agt_id2**	**agt_id3**	**agt_id4**
E_Samp	40320	40320	40320	40320
E_Send	636,016	106,010	106,010	106,010
E_LPL	8147520	8147520	8147520	8147520
E_Idel	700,85	772,85	772,85	772,85
Total	9,596,772	9,066,766	9,066,766	9,066,766

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A pair of AA batteries have approximately 2700mAh, or 9.72·109µAs: a node with this duty cycle can last approximately 2.93 years on 2 AA batteries. With no listening a node could theoretically last 28.6 years, disregarding the shelf-life of the batteries.
Agent agent_id1 consumes more energy because it sends also data of agents agent_id3 and agent_id4.

6. CONCLUSION

The contribution of this paper is to give a new formal approach to deal with specification and formal verification of a WSN. The originality is to consider each component of WSN as a Reactive Decisional Agent, and to bring together several formal synchronous modeling and validation tools. With its top-down process and its principles of decomposition, this method allows getting a model which is more easily understandable by the user. The SYNCHARTS models are used here in order to describe the reactive agent behaviors. These behaviors will be checked in a qualitative (respectively quantitative) way by the synchronous language ESTEREL (respectively by Real Time Temporal Logic deduction). The mechanism of action horizon, the time during which an agent decision remains valid, is moreover useful to specify temporal performances.

REFERENCES

[1]. Römer, Kay; Friedemann Mattern (December 2004). "The Design Space of Wireless Sensor Networks". IEEE Wireless Communications 11 (6): 54-61.
[2]. Harel, D.: Statemate: aworking environment for the development of complex systems. IEEE Software Engineering, 16(4), (1987).
[3]. Harel, D.: STATECHARTS : A Visual Formalism for Complex Systems. Science of Computer Programming, 8 pp. 231274, (1987).
[4]. Berry, G.: The ESTEREL V5 Language Primer. Internal Report, CMA Ecoles des Mine, INRIA, Paris, 17 Mars (1998).
[5]. F. Boussinot, and R. de Simone. : The ESTEREL language. Proceeding of the of the IEEE, 79(9):12931304, September (1991).
[6]. Berry, G. and P. Couronne. : Synchronous programming of reactive systems: an introduction to ESTEREL. IEEE Software Engineering, 16(4), (1987).
[7]. Ferber, J.: Les systèmes multiagents. Vers une intelligence collective. (IIA, InterEditions), 2th edn, (1997).
[8]. Nwana. H.S.: Software Agents: An overview. Knowledge Engineering Review, 11(3) pp. 205244 (1996).
[9]. R. Romadi, B.Bounabat, J.JC. Lafont et S. Labhalla., " Users's behavioral requiremnts specification for reactive agent" IEEE publication-CESA'98, Nabeul-Hammamet, Tunis, Avril 1998.
[10]. R.Romadi, B.Bounabat " Designing Multi-Agent Reactive Systems: a specification method based on Decisional Reactive Agent " PRIMA'99, 2-3 December, Japan 1999. publié au New Computing Group, February 2001.
[11]. Rahal.Romadi, Hassan.Berbia : Wireless Sensor Network. A specification method based on reactive decisional agents ICTTA08 syria 7-11 April 2008
[12]. Kevin Klues,Vlado Handziski, Chenyang Lu, Adam Wolisz,David Culler, David Gay, and Philip Levis. Integrating Concurrency Control and Energy Management in Device Drivers. SOS'07 New York 2007.