A novel design flow for physical systems simulation with sensor networks: a case study on insects distribution

Mahamadou Traore, Tran Hoang, Bernard Pottier, Ousmane Thiare

ASCII (Association Sénégalaise des Chercheurs en Informatique)

PROCEEDINGS

CNRIA

Edition 2015

6ème COLLOQUE NATIONAL SUR LA RECHERCHE EN INFORMATIQUE ET SES APPLICATIONS

du 28 au 31 octobre 2015 à l'Ecole Polytechnique de Thiès (Sénégal)

avec le soutien du CEA MITIC

Centre d'Excellence Africain en Mathématiques, Informatique et TIC

A novel design flow for physical systems simulation with sensor networks: a case study on insects distribution

Mahamadou Traoré
Université Gaston Berger
Saint-Louis, Sénégal
mahamadou.traore@ugb.edu.snl

TRAN VAN HOANG
Can Tho University
Vietnam
tvhoang@cit.ctu.edu.vn

Ousmane Thiaré
Université Gaston Berger
Saint-Louis, Sénégal
ousmane.thiare@ugb.edu.snl

Bernard Pottier
Université de Brest, LabSTICC
Brest, France
pottier@univ-brest.fr

***Abstract*—This paper presents and motivate a synthesis technique that allows to produce physical simulations jointly with network simulations. Such integration is known as Cyber-physical System. The motivation is to obtain a new design flow where observation and control networks are optimized to match status and possible evolutions of an environment model. While the control networks are expected to be mesh sensor networks, physical simulations are produced using cellular automata representing a physical state and local exchanges. Both of the simulations are time driven, synchronous and can be executed on multicore processors or graphic accelerators from the same set of tools. They can interact on sensor interfaces, possibly in realtime.**

***Keywords*—CPS, WSN, GPU, simulation**

I. Introduction

Cyber-Physical Systems (CPS) is a new concept introduced in the early 2000s. It integrates computation and environmental processes controlled and monitored by embedded systems and networks as a whole. This integration allows to collect informations from the physical environment through sensors and acting upon that environment by means of actuators.

Embedded and physical systems interaction required fundamentally new design approches because physical processes affect computations and vice versa. So, abstractions, modeling, design, and analysis techniques must concern the whole integrated system [6]. This concept is applicable to both small system such as automobiles and large system such as wide cars parking.

An example of control system with information about the physical environment available through sensors are presented in the figure I (source [8]). It shows sensors and actuators that handle analog signal values. So appropriate conversions must be performed. For that purpose, two kinds of circuits are used: sample-and-hold circuits and analog-to-digital converters (ADC). Generated results are displayed or used to control the physical environment through actuators.

Moreover, with the emergency of small radio transceivers, sensors can be associated to observe wide country parts. New applications of importances are coming, such as city and resource management, or environment monitoring. The key point is that we are moving from indeterministic situations toward deterministic observation and control systems accessible to a larger number of persons.

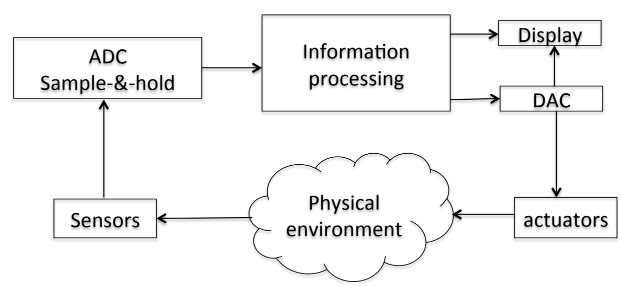

Fig. 1. Example of Cyber-Physical System

As an example , the cars parking question: I need to park my car close to some location, then I must visit around to look for a free position. Possibly for minutes, possibly for hours. There can be no position available, and many drivers can be in similar case, occasioning time and petrol wasting, and pollution of the city. At contrario, if we are able to deploy WSN for sensing each parking place and communicate status to a computing center, the solution is known immediately. Furthermore, smart phones and internet allow to obtain the conclusion and to allocate resource, saving a lot of time for drivers and citizens. Research on sensor systems did important progresses since the beginning of years 2000, with major applications beginning to appear in the last 10 years. Sensing is observing the physical reality through distributed interfaces: the case of the parking lots is well understood with some detection mechanism installed near an expected car position. A measure on a possible metallic volume each two minutes is probably enough to decide about availability of the place.

Thus, we see that a parking system can be a huge distributed sensing device sampling a physical aspect at a rate of 30 measures per hour. The device has each of its node connected using a mesh system, or other topology. To explain the situation, one can refer to a cyber-physical system having a physical aspect, and an observation/control aspect:

- the control system associates thousands of sensors, radio links, the information and reservation center, cellular phone system, car devices used for interaction (smartphone, GPS)
- The physical system groups parking positions, traffic status, and possibly other elements such as attractions or other city facts.

Another example relative to greenhouse management through the deployment of WSNs. Each WSN is composed of multiple sensors and actuators forming a climate control system with lighting, cooling, heating, carbon dioxide generating, watering, and fertilizing subsystems.Thus, light intensity, temperature, humidity, and density of carbon dioxide are collected and reported. The decision system will transform these sensing data into high-level knowledge (e.g., the proportion of each type of fertilizers) to trigger actuators to maintain good environmental factors inside [9].

A third example concerns the locusts gregarization phenomenon in gregarious area. Such phenomenon are affected by many factors such as temperature, vegetation density and rainfall. A sensor network deployment allows to receive real-time data relative to gregarization phenomenon and send an alert to a processing center. We try in this paper to build an model who integrate the wsn as control network and locusts movement impacted by such factors.

Cellular automata are used to define differents states of the whole and the simulation performed on a GPU allow to see possible evolution of the phenomenon.

This paper will present principles and tools at a geographic region level, as a number of sensible problems are appearing at this scale, notably due to the effects of climate change. We will consider wireless networks associating sensors spread over a monitored region (river, shore, forest, frontier . . .), and we will describe how these systems can interact in space and time, with a model for the environment they are managing. We consider mass-migrating locusts groups model , for this purpose, we consider for each cell a certain number of locusts, at each state a random number of locusts moves to neighborhood. We use a transition rule using random behavior, and see the system evolution. Data space could embed cell pixels (more or less vegetation), and a random number of insects.

Recent models from theoretical physics have predicted that mass-migrating animal groups may share group-level properties, irrespective of the type of animals in the group. One key prediction is that as the density of animals in the group increases, a rapid transition occurs from disordered movement of individuals within the group to highly aligned collective motion. Understanding such a transition is crucial to the control of mobile swarming insect pests such as the desert locust.

[2] confirmed the prediction of a rapid transition from disordered to ordered movement and identified a critical density for the onset of coordinated marching in locust nymphs.

II. Synchronous simulations

Time coordination is a necessity in simulation of multiple cooperating distributed systems. This coordination is a difficult since cooperation can involve very small or fast components of different behaviour. As an example a flooding appearing in a river or a shore can take hours to develop, while sensing and making decisions in the sensor system is in the scale of minutes. It is even more difficult to merge continuous simulation from mathematical models, with the passive or optimistic discrete simulation technique that we know. Coordination between concurrent simulations also involve data exchanges, as example, time reference, critical locations, measures, preliminary diagnostics about potential dangers.

Solutions for synchronizations and data exchange have been explored since the mid 1990 years producing *Discrete Event System Specification* (DEVS) methodology and *High Level Architecture* (HLA) coordination standard [12]. Both network simulators and physical simulators are intended to take place and cooperate in an real time capable, HLA framework.

A. Synchronous distributed model (SDM) and sensor networks

SDM is a well known design method for distributed algorithms where the global evolution is guided by a repeated cycle: (M_i) sending messages to neighbors and (N_i) accepting messages from neigbors, (S_i) inspect incoming messages and current state, decide about a new state (S_{i+1}), produce (M_{i+1}) messages for next cycle, then proceed to new cycle $i + 1$.

As distributed systems cannot refer to a real clocking system, this execution scheme where clock is replaced by locked steps was found to be a good method for specification and implementation of many distributed algorithms. Refering to Nancy Lynch [7], this domain can be very challenging with unknown communication topologies, unknown number of nodes, different computation speed, failure hazards, etc. . .

A fact is that this design method match perfectly sensor systems, that are highly synchronized to follow a physical sampling rate, with scheduled radio communication *rendez-vous* and sleeping periods. *Wireless Sensor Networks* (WSN) are closed system, dedicated to few tasks where distributed algorithms can be specified and implemented. Computation structure can be arbitrary in terms of process organization, and in terms of program organization, calling synchronously procedures to make distributed decisions.

B. Synchronous model and cellular automata (CA)

Embedded systems are typically reactive systems. A reactive system is one that is in continual interaction with its environment and executes at a pace determined by that environment [1]. Reactive systems can be thought of as being in a certain state, waiting for an input. For each input, they perform some computation and generate an output and a new state. Therefore, CA are very good models of such systems. Mathematical functions, which describe the problems solved by most algorithms, would be an inappropriate model.

CA is an execution model representing systems organized as cells that exchange physical elements locally. The computing model states that the whole cell systems evolves synchronously in a cycle similar to SDM: exchange data with neighbors, examine local inputs if any, decide of a new state for the cell.

As CA are defined to react to local change of states, the notion of neighborhood allow to fix some simple direct connexion topology, such as *N,W, S, E* Von Neumann neighborhood, or *N,W, S, E, NW, SW, SE, NE* Moore neighborhood. These topologies can be extended to radius greater than one, and can also be irregular as it is the case for pattern matching in image recognition. The shapes and the dimensions of a CA are also variation points with 1D or 2D representations bound to segments or rectangles, 3D suitable for geographic models, or assembly of shapes representing different cooperating behaviors.

CA are enough flexible to approximate reality in many situations going from nano organizations to galactic scale. They allow to model gaz and fluid behaviors [3], wind, snow[1], traffic and micro-to-macro scale evolutions They can reproduce higher level organizations such as rivers, roads, vascular systems, graphs[2]. Due to their inherent massive parallelism, CA is a way to address physical world modeling on best of the class computing architectures.

C. Sensors, environment control, and simulation

The sensing network is an artefact built for control and monitoring of a real physical domain. It has the ability to sense physical status, possibly several variables, possibly in many different ways. Both systems share the same physical space in which sensing location are defined. The sensor network have some support to add time stamps and location to its measures.

An example of system coupling is the *car parking application* mentioned section I, with traffic and parking being under control of a WSN. Other suitable references are seismic monitoring an analysis system[3] that reports on source of earthquakes at continent level[4], or geophysics seismic vibrators and geophone used in the *full waveform inversion* technique for underground exploration[5].

In many situations non invasive techniques are used for remote sensing: satellite, planes and unmanned aerial vehicles (UAV), underwater equipments can used several methods to obtain physical status : high resolution camera, radars, lidars, sonars are used for this In other domains, lot of health equipments also allow to obtain similar measures for different aspects of the human body, and vehicles use of sensors are in the order of 100 expected to grow to 200 by 2020. Currently, each vehicle has an average of 60-100 sensors on board. Because cars are rapidly getting smarter the number of sensors is projected to reach as many as 200 sensors per car. These numbers translate to approximately 22 billion sensors used in the automotive industry per year by 2020.

The environment basic model is certainly a spherical volume that can be conveniently divided in cells holding a geoposition such as WGS-84, Lambert, or other[6]. The nature of the application will fix cell size, and coordinates: underground/underwater, ground/ocean, air. In lot of cases a 2D representation can match needs: geolocalized maps and aerial pictures are an initial key point for space discrete division possibly followed by complementary databases inquiries. A natural organization of the physical space is thus a set of cells.

The role of a joint simulation is to mimic situations likely to happen, to check correctness of different aspects in the artefact design related to its role toward the physical domain behavior. Metrics are needed to answer a number of questions: is the deployment of sensors complete for to data collection, or are they lacks ? is it just efficient enough or surabundant ? if a natural danger occurs, such as flooding or fire, what are its chance to resist ? are the sensors installed in the best places to obtain critical information about a spurious natural risk? what will be the cost of algorithm choice in terms of energy budget ? what are the delays to obtain information on a critical physical change ?

III. Unification: everything is physical

Few basic observations allow to simplify the relative positions of the simulations.

1) the artefacts of wireless network connectivity is not very different from a sensing activity: emitting and receiving are physical facts, computing geometric characteristics of a radio is based on physical considerations: obstacles, noises, weather conditions, available power.
2) nature has its own definition of communications: rivers flow to the ocean gradually, fires spread from place to place, animals or insects reproduce and move following their development chances, pollution spread from place to place.

There is a strong chance to obtain a unified reasonning framework following simple abstractions : geometric localization, connectivity models, time references and synchronism, local behaviours and massive parallelism. Two software prototypes have been develpped to investigate this direction, NetGen for sensor network simulation, then PickCell for cellular synthesis based on 2D data presentation. Both produce efficient simulators from arbitrary topologies and arbitrary local behaviours. Simulators are concurrent programs executed on state of the art multicore processors or *General Purpose Graphic Processing Unit* (GPGPU) having thousand of processing elements (figure 2) .

IV. Related works

The authors in [12] focuses on implementation issues that arose in implementing the DEVS simulation protocol which allows for efficient event-based simulation, maximally exploit HLA primitives, and supports the full range of DEVS expressive power, which has been shown to subsume all discrete event system behaviors. Direct mapping of the DEVS protocol does not match up with the HLA specification. They discuss such a direct mapping and show where the problems arise in such an approach. Then present an alternative solution involving off-loading some of the time-management on a coordinator federate.

T. Toffoli et al in [10] recognized the importance of cellular automata as a modeling environment for physical systems. They were very interested in the analogy that exists between

[1] http://catdir.loc.gov/catdir/samples/cam031/97028284.pdf
[2] http://cdn.intechopen.com/pdfs-wm/15068.pdf
[3] http://earthquake.usgs.gov/monitoring/anss/backbone.php
[4] http://gate.iitk.ac.in/gate2012/gg.php
[5] http://en.wikipedia.org/wiki/Exploration_geophysics
[6] http://geotags.com/geo/geotags2.html

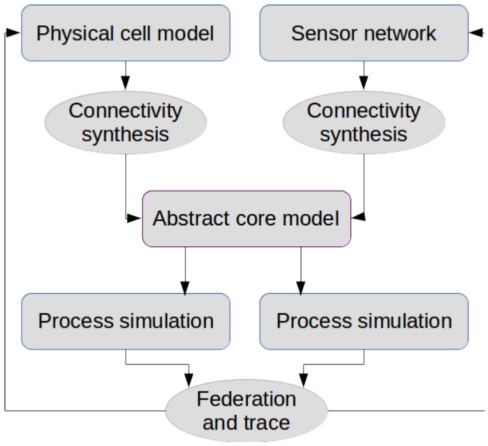

Fig. 2. General flow for physical and network high performance simulations

the theory of information as it is used to describe numerical processing in a computer and the laws of physics. Cellular automata provide an excellent framework to develop these ideas. In particular, they showed how to build a fully time reversible logic from which any numerical operation can be implemented without any loss of information. The so-called billiard ball is a cellular automata rule which is an example of such a reversible model of computation.

This work [4] has investigated the combinations of locust population density, vegetation abundance, and vegetation distribution in gregarization triggering. Locust aggregations will build into major outbreaks only if locally gregarized populations remain together and move collectively into neighboring areas of habitat, where they can recruit further locusts to the growing band. Unless such cohesive movement occurs, local aggregations will disband and individuals will return to the solitarious phase. Hence, it is vital to predict the onset of collective motion.

V. Materials and Methods

A. Case studies

With the self-propelled particles model [11] in which each particle adjust its speed and/or direction in response to near neighbor, we spread a random population of locusts on suitable zones. An CA transition rule, using random behavior is set to learn how groups form complex pattern, and observe the population evolution. We aim to simulate desert locusts aggregation as that happens in gregarious area. Gregarious area is where locusts group themselves to form a compact swarm before invasion. It's well known that the desert locusts, Schistocerca gregaria has a devasting social and economic impact on humans. Before taking flight as adults, hoppers form coordinated marching band that can extend over many kilometers. For effective management of locust outbreaks, a really control and detection of bands must be done. Because the control of flying adult swarms is costly and ineffective. The first stage in band formation is a change among resident locusts from the harmless, non-band-forming (solitarious phase), to the actively aggregating, band-forming (gregarious phase).

Many factors affect the locusts aggregation, their density, the rainfall, temperature, vegetation abundance, etc. . . .

In this work, we considered a simple model of insects aggregation with simple transition rules performing on an 2D cellular automata with Von-Neumann neighborhood. We performed simulation on an CA model where one cell contents 100 locusts, and its neighbors contains respectively for the north, south, west end east cell, 200, 10, 120 and 50 insects

At every step, considering to one cell, a number of locusts will move to its neighbor, this number is random and it will be distributed to the neighbor according to the local population of the neighbor. It means that if we have 100 locusts and randomly taking 20 locusts (20 percent of its population) for the movement. And the populations of the neighbors are 200, 10, 120, and 50 (in the case of Von Neumann 1) locusts. Then, the distribution will be 40, 2, 24, and 10 locusts respectively. In addition, some locusts regularly reproduce offspring. And the reproduction rate depends on a certain probability. Certainly, for the sake of simplicity, several other factors, which decide the movement of the locusts, are omitted such as food, wind speeds, and wind directions, the temperature, the humidity, vegetation density, etc.

B. tools

a) PickCell: Cell systems are produced using a labsticc[7] tool labs, and translated to support execution as concurrent process systems or CUDA programs. It aims to represent physical system on which acquisition and control must operate, calculate changes in this system accordingly to time, allow interaction between the control network and the physical process and generate a cellular plan as input of an Automata cellular.

The figure 3 below shows how we can segment a map, a picture or other to produce a cellular plan that can be an input for the automata cellular.

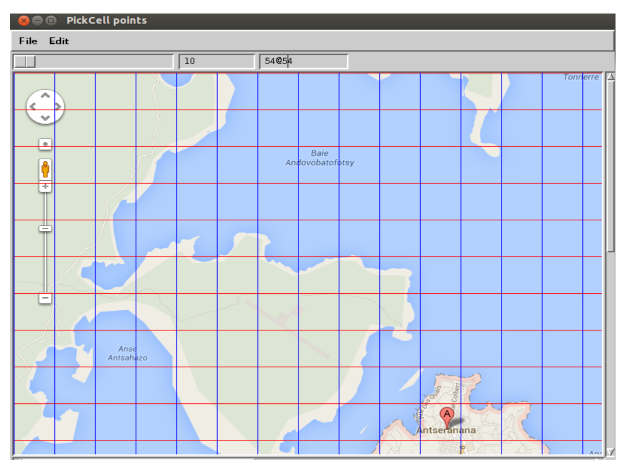

Fig. 3. Etape de segmentation

We used a NVIDIA GeForce GTX 660 Ti graphics card cuda capable with 960 cores. This card can execute hundreds of cores that can collectively run thousands of computing threads. With cuda [5]

[7]Laboratoire des sciences techniques de l'information, de la communication et de la connaissance

VI. Simulation results

The three figures below show the simulation results obtained . The figure 4 shows the first state of te CA.

Fig. 4. The CA first state

The figure 5 shows another state of the simulation. We see here that exchanges occurs and insects at time of locusts gregarization

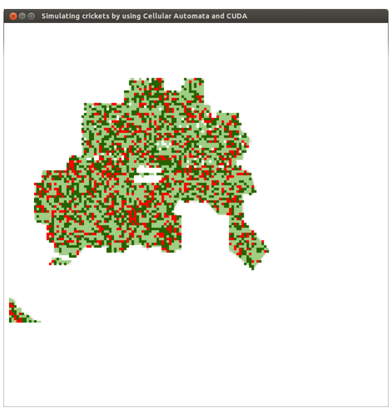

Fig. 5. a look at a CA intermediary state

The figure 6 shows the final state of the CA.

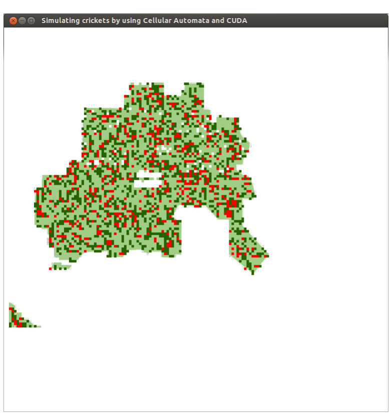

Fig. 6. The CA last state

VII. Conclusion et perspectives

In this paper, a combined technique that integrate the physical environnement model and control networks is used to simulate a locusts gregarization model on an GPU cuda-capable system . The objective was to represent a new design flow where observation and control networks are optimized to follow the evolutions of the gregarization phenomenon by using CA model. As perspective we aim to take into account others factors such as vegetation density, rainfall, temperature and humidity for approaching a real case .

References

[1] Jean-Michel Bergé, Oz Levia, and Jacques Roulliard. *High-level system modeling: specification languages*. Kluwer Academic Publishers, 1995.

[2] Jerome Buhl, David JT Sumpter, Iain D Couzin, Joe J Hale, Emma Despland, ER Miller, and Steve J Simpson. From disorder to order in marching locusts. *Science*, 312(5778):1402–1406, 2006.

[3] Shiyi Chen and Gary D Doolen. Lattice boltzmann method for fluid flows. *Annual review of fluid mechanics*, 30(1):329–364, 1998.

[4] Matthew Collett, Emma Despland, Stephen J Simpson, and David C Krakauer. Spatial scales of desert locust gregarization. *Proceedings of the National Academy of Sciences*, 95(22):13052–13055, 1998.

[5] A. Iqbal and B. Pottier. Meta-simulation of large wsn on multi-core computers. In *DEVS10, Spring Simulation Multi-Conference, Orlando, SCS*, Apr 2010.

[6] Edward A Lee. Cyber-physical systems-are computing foundations adequate. In *Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*, volume 2. Citeseer, 2006.

[7] Nancy A Lynch. *Distributed algorithms*. Morgan Kaufmann, 1996.

[8] Peter Marwedel. *Embedded system design: Embedded systems foundations of cyber-physical systems*. Springer Science & Business Media, 2010.

[9] V.L. Narasimhan, A.A. Arvind, and K. Bever. Greenhouse asset management using wireless sensor-actor networks. In *Mobile Ubiquitous Computing, Systems, Services and Technologies, 2007. UBICOMM '07. International Conference on*, pages 9–14, Nov 2007.

[10] Tommaso Toffoli and Norman Margolus. *Cellular automata machines: a new environment for modeling*. MIT press, 1987.

[11] Tamás Vicsek, András Czirók, Eshel Ben-Jacob, Inon Cohen, and Ofer Shochet. Novel type of phase transition in a system of self-driven particles. *Phys. Rev. Lett.*, 75:1226–1229, Aug 1995.

[12] Bernard P Zeigler, George Ball, Hyup Cho, JS Lee, and Hessam Sarjoughian. Implementation of the devs formalism over the hla/rti: Problems and solutions. In *Simulation Interoperation Workshop (SIW)*, number 99S-SIW, page 065, 1999.