



## Specifications Table

Subject area	Computer Science
More specific subject area	Agent Simulation and Modeling
Method name	Agent Oriented Software Engineering/ Agent Oriented Analysis and Design Method
M. Wooldridge, P. Ciancarini, Agent-oriented software engineering: The state of the art, in: Agent-oriented software engineering. Springer Berlin Heidelberg, 1–28, 2001	Summarily, they asserted that the major roles played by formal methods in software engineering include; the specification of systems; for directly programming systems; and in the verification of systems. This implies that specification, programming and model verification are the stages of their methodology.

## Method details

In computational modeling, most researchers represent and illustrate dynamical systems such as networks in equilibrium (or steady state) or going between steady states. However, agent modeling (a form of individual based models (IBMs)) can complementarily add to the equation-based approaches. Where equation-based approaches allow researchers to characterize the steady states of a dynamical system, multiagent modeling permits the strong likelihood and practicability of producing those steady states. Perhaps the productive and supplementary aspect constitutes the major advantage of complementing analytical modeling with agent-based modeling. This is because agent modeling employs simple rules that result in diverse complex behaviors of a real world phenomenon. Since emergence and complexity are its essential concepts, agent modeling goes beyond the steady state-orientation of analytical modeling to the investigation of a system's robustness and adaptability.

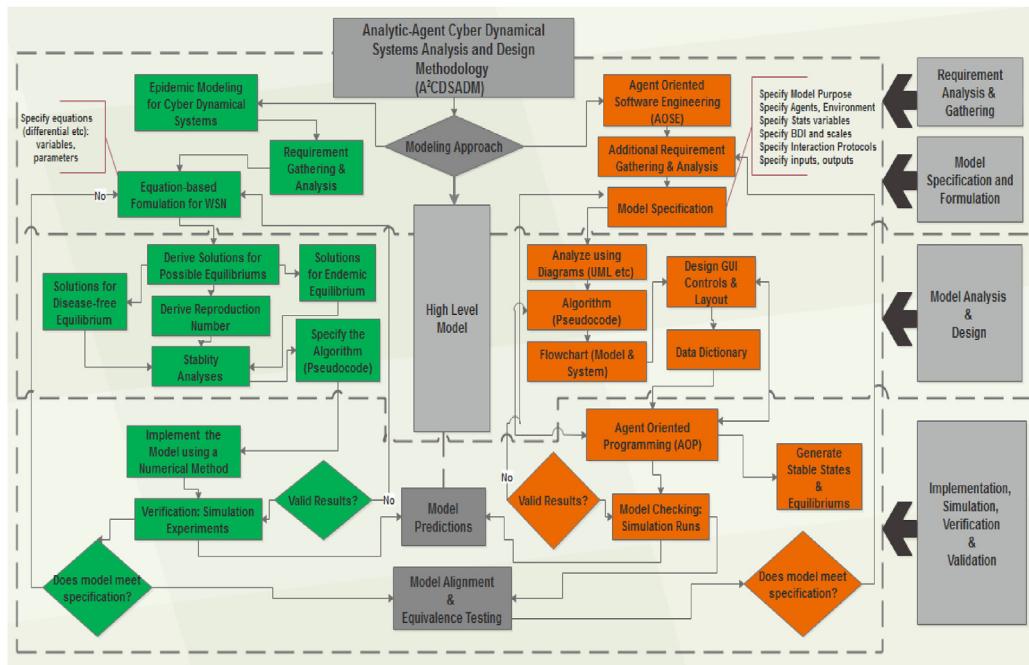


Fig. 1. Diagrammatic representation of the A<sup>2</sup>CDSADM.

However, there is no formalized method for achieving the productive and supplementary contribution of agent-based modeling to analytical modeling methods.

For the equation-based (analytical) approach we adopt a method gleaned from the extant literature on network epidemiology (which we referred to as) the *Epidemic Modeling and Analysis of Cyber Dynamical Systems*. For the Agent-based approach we employ the *Agent Oriented Software Engineering* approach [1] which applies the *Agent Oriented Programming* (AOP) in its implementation. Merging the two approaches is very necessary in order to employ their respective strengths. This merger resulted to a new method, referred to as the ***Analytic-Agent Cyber Dynamical Systems Analysis and Design Method (A<sup>2</sup>CDSADM)*** (Fig. 1). While analytical modeling achieve the development of traditional analytical WSN models by compartmentalization of nodes according to their health status, agent modeling would enable the building of agent simulators with more spatial capabilities using simple rules. The developed simulator would go beyond representing some characteristics of our proposed models to easily characterizing factors such as mobility, packet transmission, multi-group malicious code transmission, carrying capacity, sensor monitoring, sensor (daily) ageing, life span and death etc.

Essentially, the method complements the creation of accurate agent based models with a benchmark analytical model (BAM) involving a system of ordinary differential equations. This BAM aids foundational validation of the first agent model before subsequent modification. Note that the creation of a BAM became necessary because it was found that in some studies that attempt to validate agent models of malware propagation with an analytical model, the obsolete SIR model [2] was used; such works include [3–10]. One of the reasons for doing so is the well known nature of the SIR model; another reason might be due to the unavailability of an equivalent analytical model. This is not only inadequate but inappropriate because the SIR model does not involve other complexities that may exist in a real world epidemic scenario. In our study for example, the SIR model does not include the dynamical behaviour of the exposed and the vaccinated compartment as seen in the SEIR-V model.

The well known nature of the SIR model is based on simplicity, clarity, ease of implementation [4] and proven accuracy. However, these advantages could be achieved using the proposed method. Specifically, an analytical model of interest could be modified and validated before it is used to validate the resulting agent based model. On the question of validation, note that the essentiality of A<sup>2</sup>CDSADM becomes clearer and more vivid in cases where there is the lack of real world (field) data or the lack of real geographical locations of the occurrence of particular cases. The analyst/modeler may resort to some sort of cross-validation; to develop a richer model alongside a general model i.e. creating a benchmark with the traditional analytical model which will be used for comparing and validating the agent model. This type of validation (in A<sup>2</sup>CDSADM) is not only helpful, but verily expedient if; 1. there exists no analytical model that closely mimics the proposed agent model and 2. part of the modeler's objectives is the incremental modification of the agent model (simulator) to mimic real world WSN scenarios (the additional information section addressed validation processes and its attendant issues).

The new method combines the strength/benefits as well as the essential aspects of the two methods. Essential aspects imply those steps that allow the achievement of the complimentary and generative role of agent-based method to the traditional analytical approach as well as other software engineering approaches that will enhance modification and reproducibility of resulting models.

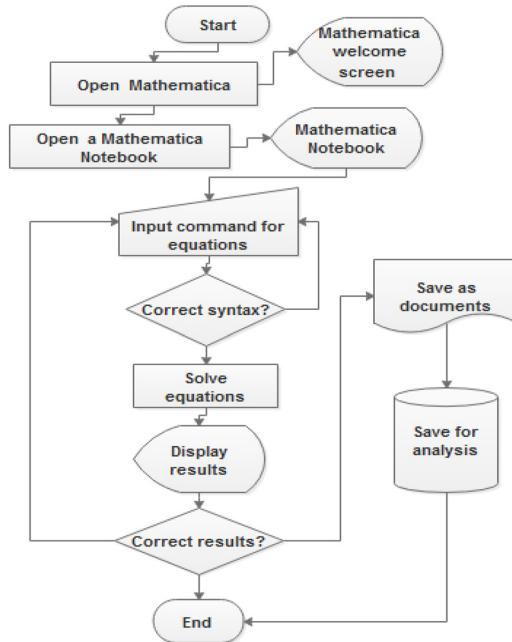
We chose the differential equation modeling method as the analytical approach for our new method because they are popular and they dominate the literature on malware propagation. To buttress this viewpoint, Martín [11] posits that, “ . . . most of the mathematical models designed to study the propagation of malware are based on the use of differential equations”. Our decision to use a differential equation modeling method (gleaned from literature) was founded on this rationale. On the other hand, we choose agent modeling because it offers more benefits than the cellular automata approach – another form of individual based models. Individual-based models are briefly reviewed at the last section of the paper.

From the graphical abstract, the methodology is vertically divided into; 1. Requirement Gathering and Analysis; 2. Model Specification and Formulation; 3. Model Analyses and Design and 4. Implementation: Simulation, Verification and Validation. To use the method one can perform the analytical modeling first and then perform the agent modeling before generating predictions and model alignment. On the hand one can perform all the activities that constitute each vertical division (for both modeling approaches), then finally perform model predictions and model alignment.









**Fig. 3.** Procedure for using a Solver.

Diekmann *et al.* [28] and the inverse of the susceptible class at the endemic equilibrium used by Mishra & Pandey [16]. The hypothetical study used employed the latter instead of the former; therefore the Reproduction number is;

$$R_0 = \frac{\beta\theta\sigma\pi r_0^2}{(\theta + \tau)(\nu + \tau + \omega)} \quad (4)$$

Here  $R_0$  is a mathematical expression which involves infection capacity, rate at which exposed nodes become infectious, the effective contact rate (communication range, density) for transfer of infection, rate of recovery, death rate as a result of hardware/software failure and worm infection. The condition for the existence of the endemic equilibrium goes thus: At  $R_0 < 1$ , the infection dies out and WSN is free of the malicious code, but at  $R_0 \geq 1$  the worm infection spreads and an epidemic occurs in the network.

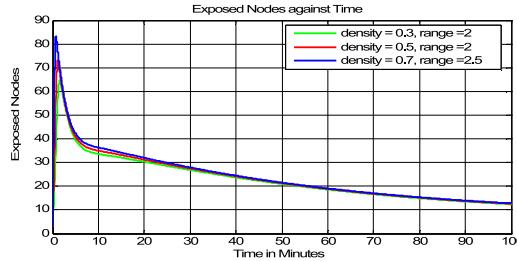
In case of comparative studies using epidemic models, the analyst can (aside stating the new reproduction number), also state the reasons behind any obvious similarities/differences when compared with the model that motivated the current study. For instance, since the epidemic threshold of the case study have been stated, it is clear that it involves parameters for distribution density and transmission range which is absent in the epidemic threshold of [12]. The epidemic threshold of [12] is;

$$R_0 = \frac{\alpha\beta}{(\mu + \alpha)(\mu + \varepsilon + \gamma)} \quad (5)$$

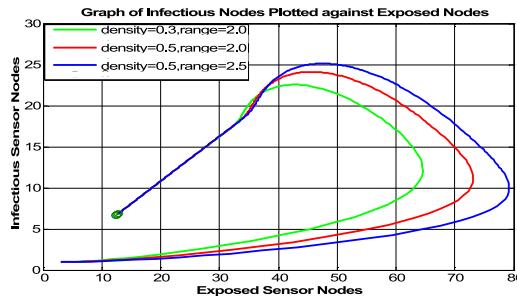
In addition, the two reproduction numbers were derived using different approaches yet they gave similar result excepting the newly added parameters for density and range. Specifically, while (5) was derived using the next generation matrix method, (4) was derived by finding the inverse of the susceptible class at the endemic equilibrium. This also shows to a large extent that the reproduction number can be validated by using one approach and checking its accuracy using the other.







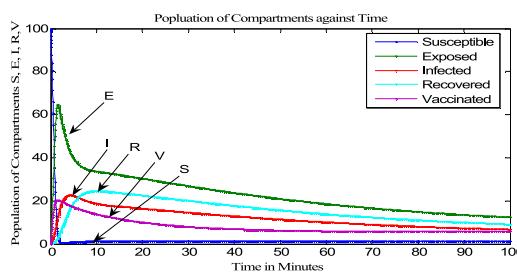
**Fig. 5.** Time History w.r.t. to  $\sigma$  and  $r_0^2$ .



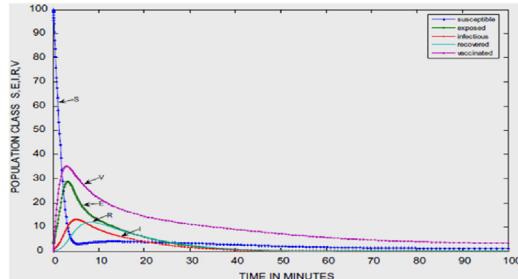
**Fig. 6.** Infectious vs Exposed Class w.r.t. to  $\sigma$  and  $r_0^2$ .

### Agent-based modeling (ABM) of the SEIR-V model

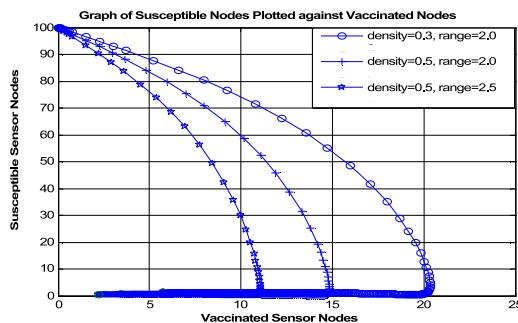
Agent Oriented Software Engineering (AOSE) – has become one of the active areas of computing research because it involves the creation of software agents or multi-agents bearing in mind concepts such as autonomy, social ability and agent interaction. The steps to the successful application of AOSE have been referred to as *Agent Oriented Analysis and Design* [1]. Wooldridge & Ciancarini [1] asserted that the major roles played by formal methods in software engineering include; the *specification* of systems; for *directly programming systems*; and in the *verification* of systems. These roles encapsulate the core thematic steps of Agent Oriented Analysis and Design or what Jennings & Wooldridge [31] called the Agent Oriented Life Cycle. The concepts of what these roles mean for an ABM or MAS, alongside some addition of steps that ensures the reproducibility of resulting models are presented below.



**Fig. 7.** Time History of [13].

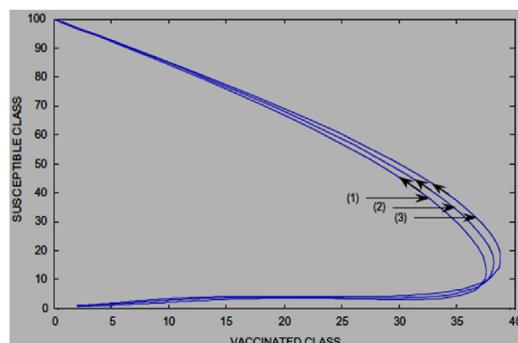


**Fig. 8.** Time History of [12].



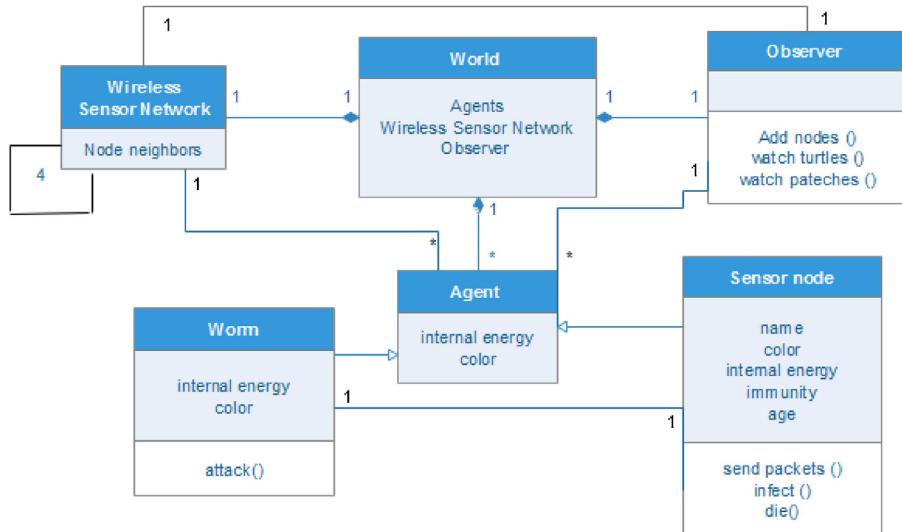
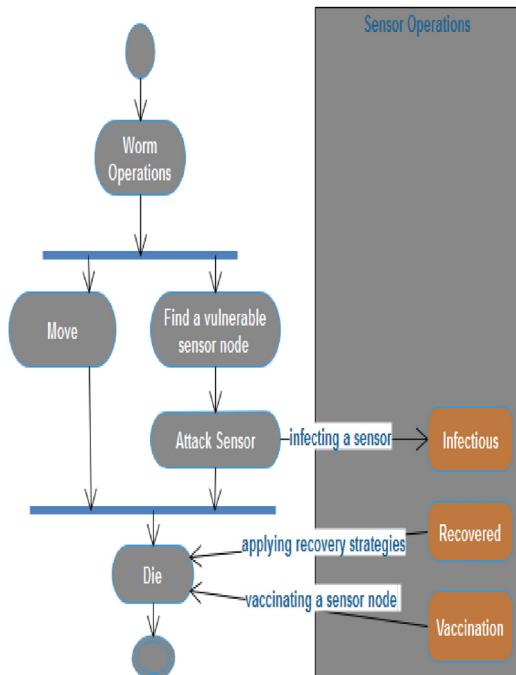
**Fig. 9.** Susceptible vs Vaccinated w.r.t. to  $\sigma$  and  $r_o^2$  [13].

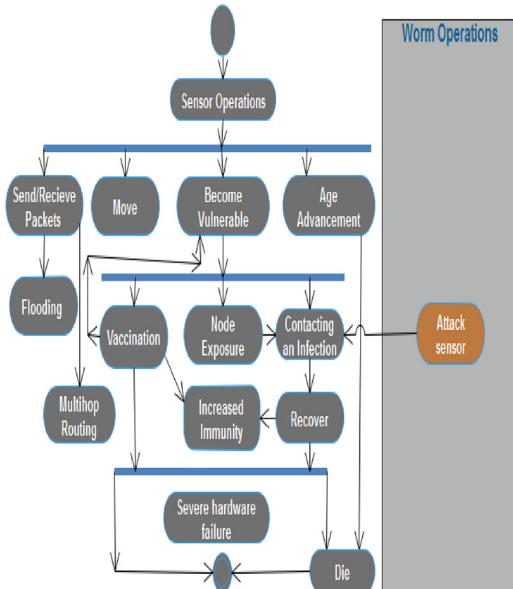
With the agent based modeling approach, the aim is to *reify* thus *embody* the factors of malware propagation in WSN i.e. building a Sensor Worm Spread Simulator (SWSS) that goes beyond basic mathematical principles to a richer representation of real-world WSN scenarios using the NetLogo agent toolkit. We intend to incrementally modify the developed agent model; and the basis for excellent modification would be justified at first using the already built analytical model (BAM). The following steps constitute the agent modeling approach;



**Fig. 10.** Susceptible vs Vaccinated Class [12].



**Fig. 11.** Class UML Diagram.**Fig. 12.** Activity UML Diagram for Worm Operation.



**Fig. 13.** Activity UML Diagram for Sensor Operation.

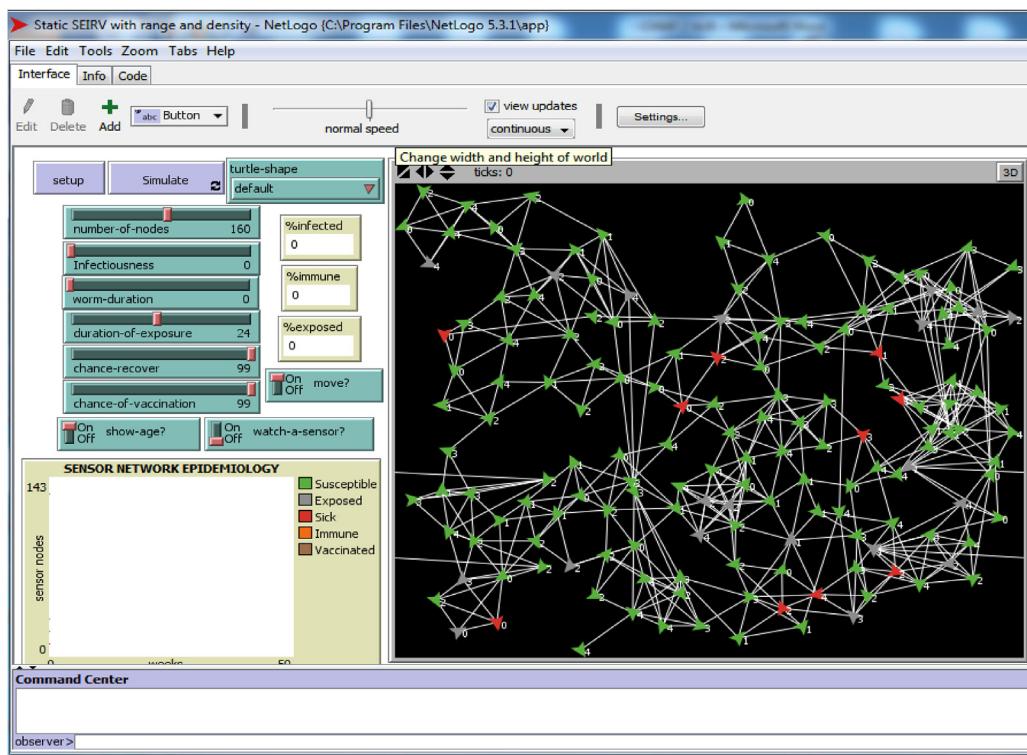
made to control events therein. Additionally, it can also display work flows and processes in the system. It is evident from the system flowchart below that the user starts the simulation model (or simulator). Then the model parameters of attributes are set in the light of what the user hopes to achieve when the simulator is displayed. Simulating either packet transmission or worm propagation here implies running several underground NetLogo instructions and displaying results as plots. The NetLogo BehaviourSpace tool can be used to run several simulation experiments which are collated, sorted and saved in a database.

Including UML and algorithm (either as pseudocodes or flowcharts) in agent based modeling aids faster and accurate design of the GUI controls. Additionally, flowcharts (and activity UML diagrams) are essential, “if one would want to deal with the more procedural instructions flow related parts of the codes” [39]. Furthermore, it aids easy modification of the models at subsequent times; and makes the resulting model reproducible. Fig. 14(a) depicts the flowchart of the proposed agent model while Fig. 14(b) depicts the system flowchart for using NetLogo agent toolkit.

#### Pseudo-code for the Sensor Worm Spread Simulator

- 1 Declare local variables for the turtles (sensor nodes and worms)
- 2 Declare global variables for the turtles (sensor nodes and worms), patches and the observer
- 3 Set up the procedures
- 4 Set up the sensor nodes
- 5 Set up sensor nodes in a wireless sensor network
- 6 Declare the procedure for a sensor node to be sick and infectious
- 7 Declare the procedure for a sensor node to be healthy and vulnerable
- 8 Declare the procedure for a sensor node to recover and become-immune
- 9 Declare the go procedures for the sensor nodes
  - a Run procedure to make the sensors nodes to get older, to recover-or-die if sick?, to infect, if else sick?
  - b Run procedure to add more sensors, to update global variable, to update display and to progress tick
- 10 Update-global-variables





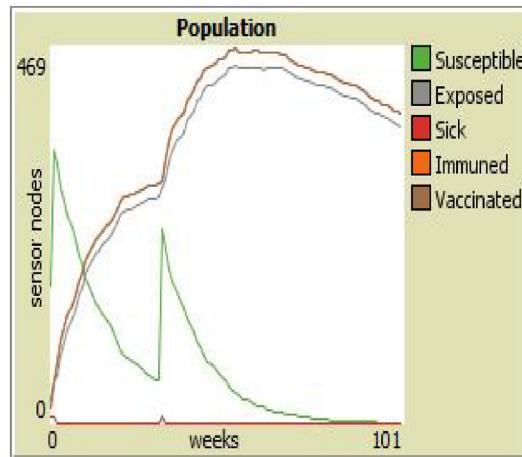
**Fig. 15.** GUI Control Center for the Agent Model.

### Implementation

At this stage the modeler goes beyond abstract specification to concrete implementation (i.e. a computational system). Executing or animating the specification may involve some level of programming – *agent oriented programming (AOP)*. This term was coined by [40]; and used to describe a specialization of (OOP). He maintained that, “AOP specializes the framework by fixing the state (now called *mental state*) of the modules (now called *agents*) to consist of components such as beliefs

**Table 2**  
Data Dictionary for Sensor Worm Spread Simulator.

Attribute Name	Required	Widget	Length	Description
Number-of-nodes	Yes	Slider	300	This controls the number of sensors in the wireless sensor network.
Design Network Setup	Yes	Button	N/A	This allows the deployment and linking of sensor nodes to form a wireless sensor network during packet transmission.
Show-age?	Yes	Switch	On/Off	A switch that allows or disallows the display of a sensor's age. Note that a sensor dies whenever its age gets bigger than the stipulated life span of a sensor.
%Vaccinated	Yes	Monitor	100	Shows the percentage of the vaccinated sensor nodes from the total sensor population.



**Fig. 16.** Infectiousness and Worm duration is 0.

(including beliefs about the world, about themselves, and about one another), capabilities, and decisions, each of which enjoys a precisely defined syntax".

Coding the agent model should be done alongside the design of the graphical user interface (GUI) controls/layout and data dictionary development. At this juncture the modeler writes rules/instructions (according to the syntax of the toolkit) that animate the earlier specifications made.

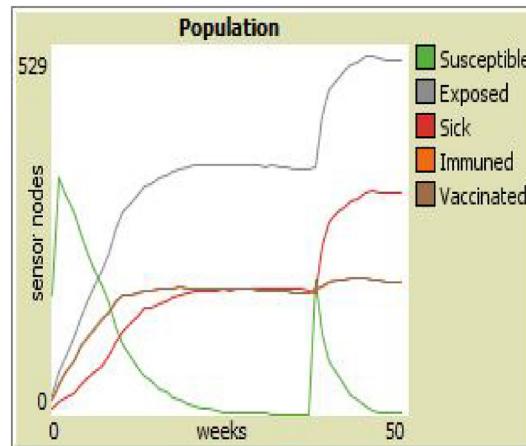
#### Verification and validation

There is need to show that this developed system is accurate in the light our original specification. This process is known as *verification*, and it is specifically essential for the development process. Jennings et al, the stage is divided into two broad classes: (a) *axiomatic*; and (b) *semantic* (model checking) and since, axiomatic methods possess some limitations (syntactic proof problems), model checking is preferably used because it is based on the semantics of the specification language. The developed models are tested, and used to make predictions. Fig. 16 shows the result of the agent model when Infectiousness and Worm duration is 0, while Fig. 17 shows the result of the agent model Infectiousness and Worm duration is 99.

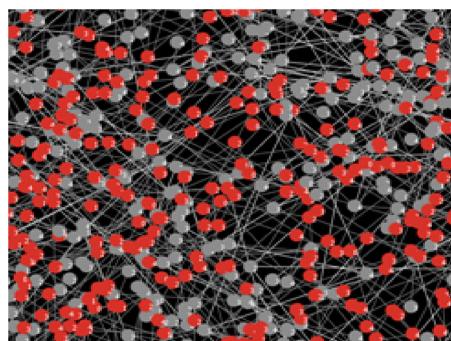
#### Equilibrium generation

With the help of A<sup>2</sup>CDSADM, visual essences of our analytical equilibrium solutions are generated using the simulator (or agent model). These visual essences/equivalents will not only depict the health status of the sensor nodes (using colors) but show possible spatial orientation of the sensors in the WSN environment.

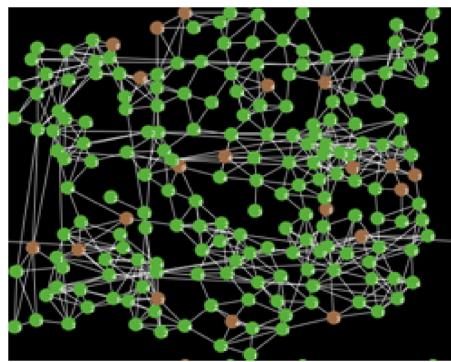
Given the colors of different sensor nodes in Sensor Worm Spread Simulator (SWSS), Fig. 18 shows the endemic equilibrium in the wireless sensor network environment. This is a point in which most sensor nodes carry the infection. The grey nodes (which are the exposed sensor nodes) will soon turn red and be ready to pass the infection to other susceptible nodes if any is deployed. On the other hand, Fig. 19 shows a malware-free equilibrium in the network environment. This is a point in which most sensor nodes do not carry the infection (i.e. most nodes are either susceptible or vaccinated). Fig. 19 is what most organization that use wireless sensor network strive to achieve for the achievement of their meaningful daily activities.



**Fig. 17.** Infectiousness and Worm duration is 99.



**Fig. 18.** Endemic Equilibrium for SWSS.



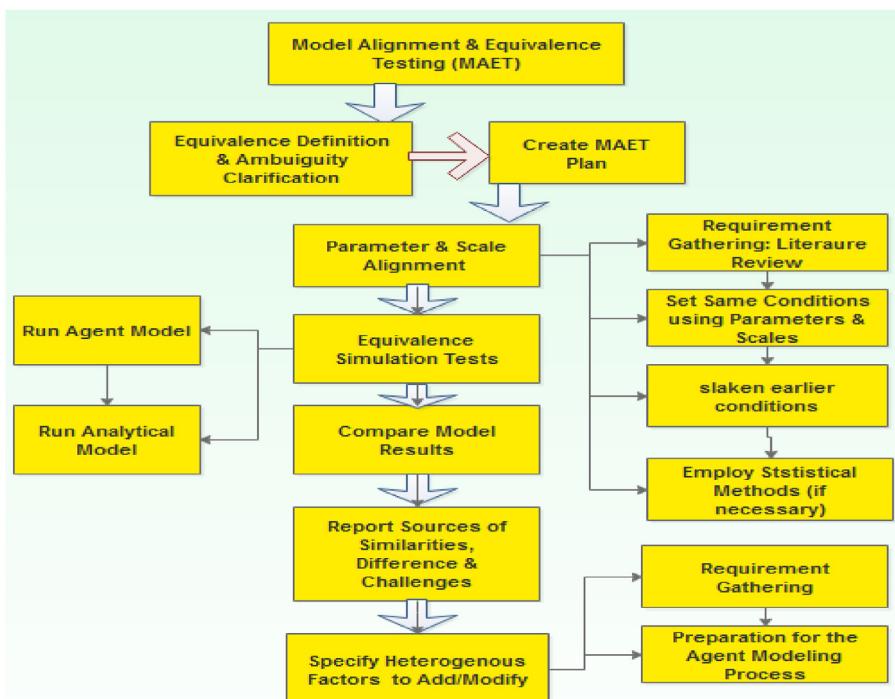
**Fig. 19.** Malware-free Equilibrium for SWSS.

### Model alignment and equivalence testing (MAET)

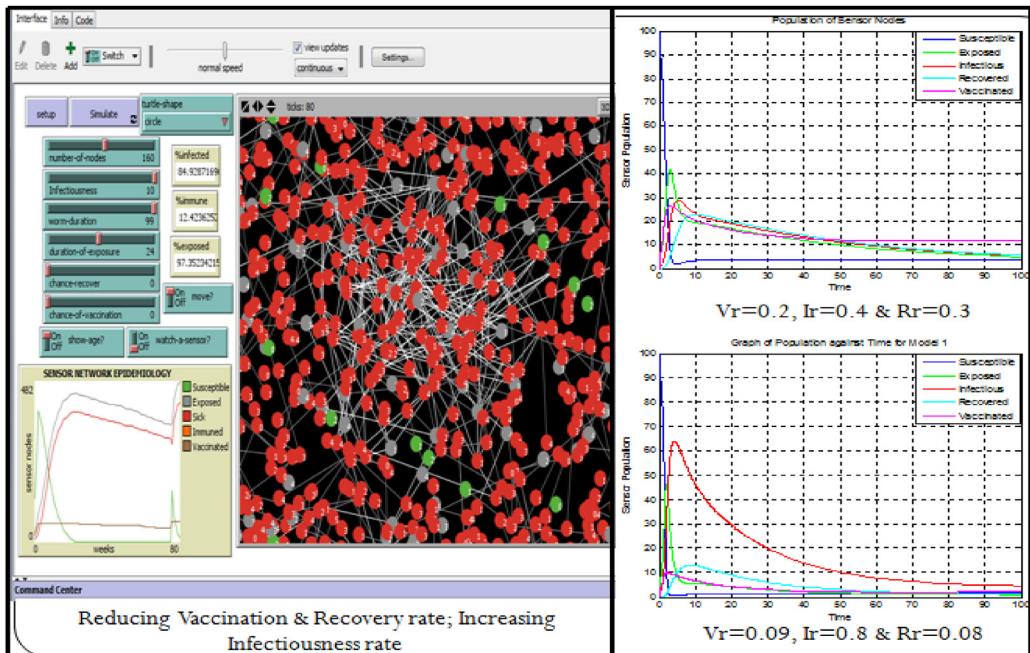
Model alignment is an evaluation approach, and as [3] puts it, “it involves aligning both computational models or ‘docking’ to match the output of the proposed agent-based models to that of the epidemic models using available information about the malicious-code spread characteristics”. The essentiality of model alignment and equivalent testing is captured in two words, “critical experiment and subsumption” [41]. If two models (with distinctively different modeling approaches) attest to representing the same phenomena, then it is fundamental to know whether they can/cannot generate same results. This is the rationale for performing “critical experiments and for tests of whether one model can subsume another” [41]. Successful MAET process encourages justified “subsumption” i.e. rejecting the earlier (analytical) model in favor of the later (agent) model that (perhaps) has a richer representation or saying the later model is a significantly exceptional expression of a traditional one.

From the graphical abstract, model alignment and equivalent testing may seem like the final stage of the A<sup>2</sup>CDSADM. However, if the analyst/modeler wants to incrementally modify the agent model to a richer and fuller realization of WSN epidemiology, then it is not. Simulation results are compared, to gauge the correctness of the agent model (or simulator) by validating it against the analytical SEIR-V model. Although this may not be entirely sufficient, but achieving (result equivalence) reflects some sort of convincing plausibility for the simulator and establishes confidence that may lead to its further modification and validation.

**Fig. 20** describes a formalized MAET process for A<sup>2</sup>CDSADM that aids easier modification of the agent model. The MDET process in **Fig. 20** consists of seven major steps; 1. *Equivalence Definition and Ambiguity Clarification*. 2. *Create a MAET plan*. 3. *Parameter and Scale Alignment*: this stage may include literature review, setting same conditions in accordance to review details, slackening the earlier



**Fig. 20.** MAET process for A<sup>2</sup>CDSADM.



**Fig. 21.** Model Alignment using the Results of Both the Simulator and the Analytical Model.

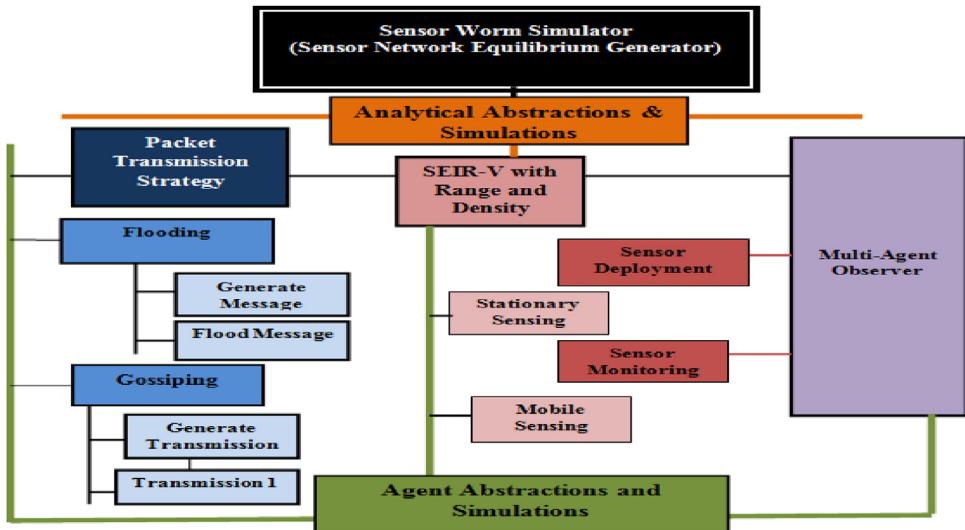
conditions and statistical activities for equivalence determination. 4. *Equivalence Simulation Tests*: this is the actual running of the two models. 5. *Compare Model Results*. 6. *Report Reasons/Sources of Similarities, Differences, Challenges*. 7. *Specify more Heterogeneous Factors to Add/Modify*: this stage would involve requirement gathering in preparation for going through the modeling process again.

Specifically, we simulated the effect of reducing the vaccination and recovery rate; and increasing the infectiousness rate. On the right hand side (of Fig. 21) is the result of using the Sensor Worm Spread Simulator (the agent-based model); while on the left hand side (of Fig. 21) is the result of performing the above operation using the analytical SEIR-V model. On the simulator, we set the “Infectiousness” and “worm-duration” sliders to 99 and setting the “chance-recover” and the “chance-vaccination” sliders to 0; correspondingly reduced the number of susceptible and increased the number of infectious (sick) sensor nodes in the network environment. Specifically, on the analytical model the vaccination rate was reduced from 0.2 to 0.09; the recovery rate was reduced from 0.3 to 0.08 and infectiousness rate was increased from 0.4 to 0.8.

To some extent the (curves) results from both approaches are approximately the same, at least in the increase in the infectious nodes and the reduction of the susceptible nodes. On the plot of the simulator (at the right hand side of Fig. 21), the susceptible sensor nodes was seen to rise again at the 80<sup>th</sup> week. This is due to the deployment of a new set of sensor nodes when the number of sensors goes below the carrying capacity of the WSN environment. Subtle differences in the results may be related to stochasticity and heterogeneous mixing of agents in the WSN.

## Additional information

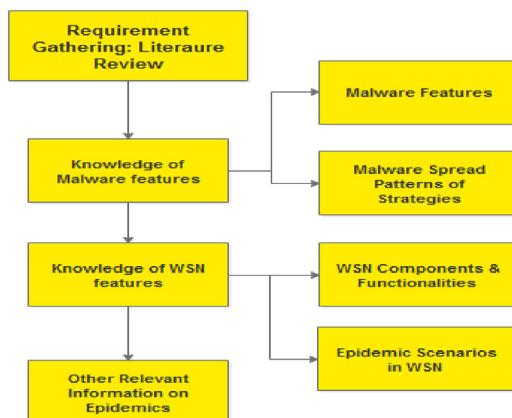
The analyst/modeler may decides to approach the A<sup>2</sup>CDSADM using the vertical division view; this would entail performing the activities in each of these stages, namely; *Requirement Gathering and Analysis*; 2. *Model Specification and Formulation*; 3. *Model Analyses and Design* and 4. *Implementation*:



**Fig. 22.** High Level Model for WSN.

*Simulation, Verification and Validation.* However, in between the two modeling approaches is the high level model (Fig. 22). Alongside performing requirement gathering and analysis, the analyst should create the high level model. This is a unified conceptual model that describes the features to be represented/characterized using both approaches of the A<sup>2</sup>CDSADM. In view of our hypothetical case studies, Fig. 22 represents the high level model for WSN. Note that the high level model should be updated as modifications are made to the resulting agent model as per the addition of more features.

Requirement gathering and analysis is a process of sourcing, reviewing and understanding pertinent literature of a real world phenomenon in order to identify the factors/characteristics to be included during model specification and formulation. To accurately perform epidemic studies in WSN, the requirement gathering and analysis stage must elicit firstly the malware characteristics and its spread patterns/strategies; and secondly the WSN characteristics/components as well as other



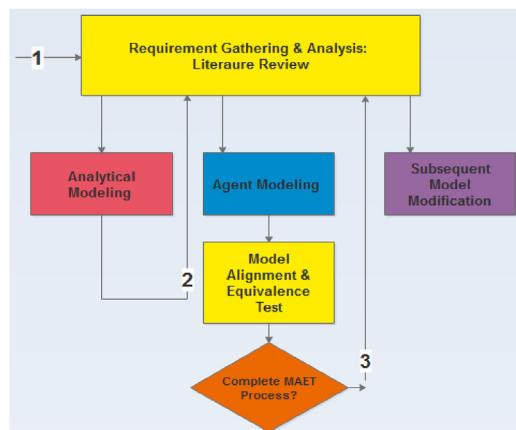
**Fig. 23.** Requirement Gathering in a WSN Context.

relevant information on epidemics (as shown in Fig. 23). Then in the light of the analyst's mathematical ability, these characteristics are formulated using equations. The analyst should thoroughly tinker on the generated information so as to ensure that only the relevant features are represented; this is the rationale for including "analysis" in the requirement gathering process. With proper requirement analysis and gathering, the method can be used to model epidemics in other networks.

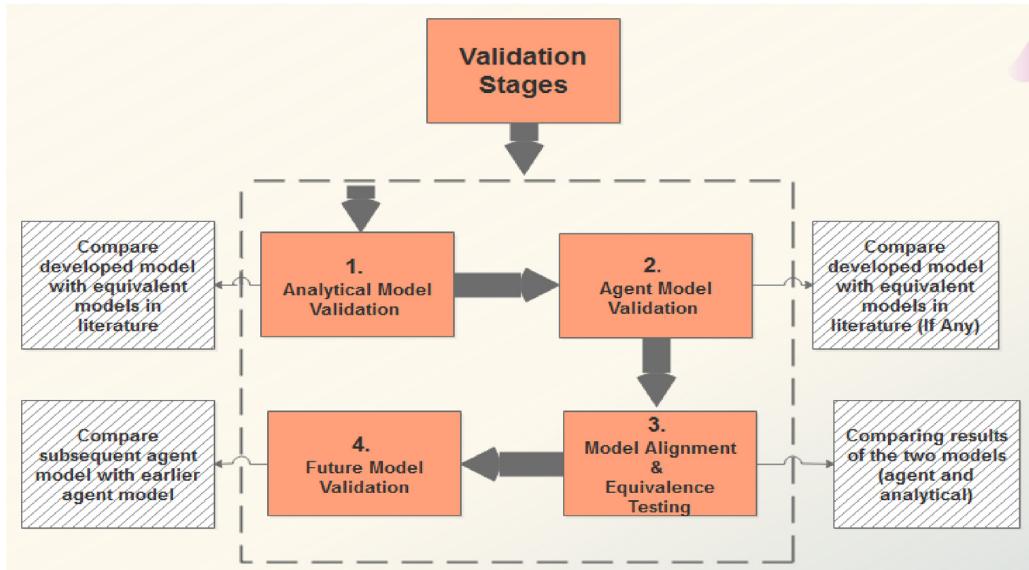
Since, we intend to achieve the entire accurate representation of WSN features and scenarios, requirement gathering becomes an iterative process that begins before model formulation and may not end, if one intends to incrementally modify the agent model. The process ends at the point the analyst/modeler decides that the resulting model can be used for accurate decision making on issues related to the modeled phenomena. Fig. 24 shows the iterative process of requirement gathering and analysis. Arrows 1, 2 and 3 pointing upwards depicts the times when the analyst/modeler performs the activities depicted in Fig. 23; while the arrows pointing downward signifies the series of activities that constitute the modeling approaches within A<sup>2</sup>CDSADM.

Model validation in the A<sup>2</sup>CDSADM, is a continuous task that aims at generating an accurate model. The points of validation are labeled with numbers from 1 to 4 in Fig. 25. Assuming the analyst does not want to modify the agent model further, the validation ends at stage three; otherwise the 4<sup>th</sup> stage validation stage is performed. The validation stages include: 1. Analytical Model Validation: compare developed analytical model with equivalent analytical models in literature; 2. Agent Model Validation: compare developed agent model with equivalent agent models in literature (if any); 3. Model Alignment & Equivalence Testing (MAET): compare results of the two models (agent and analytical) and 4. Future Model Validation: compare subsequent agent model with earlier agent model. Our validation ended at the stage 3, this is due to the absence of an equivalent agent model in literature. Note also that stage 2 (i.e. Agent Model Validation) was omitted. After MAET, the future modification of the agent model necessitates the 4<sup>th</sup> stage of validation.

Beyond the stochasticity that may be achieved by the agent model specifications, epidemic studies can be extended to involve the heterogeneity observable in a real world WSN. The heterogeneity can be in different sizes for the sensors, where a larger sensor node signifies more battery power and smaller sensor node signifies less battery power. The sensors closer to the base stations can have more battery power/energy as they are the cluster heads. Additionally, sensor networks can also be heterogeneous in terms of computational capability of the sensor nodes, bandwidth capability of the links and the initial energy of the sensor nodes. The sensor nodes can be made to send and receive packets



**Fig. 24.** Iterative Process of Requirement Gathering.



**Fig. 25.** Validation Stages for A<sup>2</sup>CDSADM.

Depending on the intended objectives, an analyst can go beyond the specifications made herein to include factors of cognition, self-organization and cooperation in the attributes of the developed agent based model. More so, adapting the OOD protocol to the A<sup>2</sup>CDSADM by adding parameters for learning, adaptation and prediction (if necessary). Additionally, the analyst can also specify agent roles or protocols of interactions between the agents or between the agent and the environment. Note that in comparative epidemic studies using analytical models, the output of any stage of vertically approaching A<sup>2</sup>CDSADM are compared to its equivalent(s) in literature.

### Malware propagation using individual-based models: state of the art

In order to understand malware spread in telecommunication/technological networks equations-based models have been used to characterize the dynamics of interaction. These models in the light of their characteristics are basically seen in literature as deterministic/stochastic, continuous/discrete and global/individual models etc. The bulk of available models are deterministic/stochastic and continuous/discrete and they pursue the global perspective of evolution wherein the overall population dynamics are investigated. From recent studies [11,42], it has become very clear that malware propagation models based on differential equations are plagued with several shortcomings and drawbacks, irrespective of its successes and popularity in both biological and network epidemiology. The noted drawbacks include homogenous mixing and distribution, inability to represent individual dynamical behavior and the inability to account for local infections between nodes in a network [11]. In other words, “models based on differential equations fail to capture the local characteristics of spreading processes, nor do they include interaction behaviors among individuals” [43].

Researchers suggested the use of Individual based models (IBMs) in order to salvage the above shortcomings. IBMs attempts to highlight the real-world autonomy of interacting individualshosts. Studies involving IBMs simulate local interactions between cells/agents in discrete time and space so as to produce emergent outcomes. Examples of IBM are cellular automata (CA) and agent based models (ABM) [42,43]. Few of these models exist for malware propagation, perhaps due to the complexity of representing individual level mechanisms of a particular phenomenon. Though both CA

and ABMs model individual level representations, their emphasis is a little different. While CA focuses on emergent outcomes of local interactions, in ABM agents take actions based on their locally coded individual behaviours and that of the environment [44]. Perhaps, more advantage lies with ABM obviously due to the complex representation of explicit behavioral processes.

CA is a discrete, deterministic mathematical model, where space, time, and the state of the system are distinct [43]. It was used to model malware propagation in wireless sensor networks [45] and on smart phones/mobile devices [43,46–49]. Using ABM, Bose & Shin [50] built a framework for malware spread in a heterogeneous environment while Hosseini et al. [51] modeled the outbreak of malware based on the rumor diffusion process.

## Conclusion

Motivated by the sameness in connectivity realities between biological viruses and malware equivalents, we sought to study the spread of malicious codes in close-to-real world WSN environment. Due to the limitation of the traditional analytical model which becomes less tractable with the addition of spatial features in WSN epidemiology, we complimentarily designed equivalent agent model that at first provided visualization and spatial orientation for the deployed sensor nodes. Furthermore, it reified other factors such as carrying capacity, sensor mobility, packet transmission etc.

These models served as hypothetical cases for merging the benefits of two distinctively different modeling approaches that resulted to a hybrid method, coined as the *Analytically-Agent Cyber Dynamical Systems Analysis and Design Method (A<sup>2</sup>CDSADM)*. This method brings more to the table; it combines the modeling of cyber dynamical (defense) systems (i.e. networks) with Agent Oriented Analysis and Design (AOAD) – an extension of the Object Oriented Analysis and Design (OOAD)/Object Oriented Software Engineering. Its coding/implementation required the application of new perspective in software engineering called the Agent Oriented Programming (AOP) – an extension of the Object Oriented Programming (OOP).

The methods that constitute A<sup>2</sup>CDSADM were extended to cover the additional features for generating the (analytical) equilibriums of the modelled system (i.e. wireless sensor network), for ensuring model specification accuracy by performing model alignment and equivalence tests, for creating a high level conceptual model containing the envisaged WSN features to be represented, for representing more complex factors of a real world WSN scenario; and to thereafter compare the result of two models.

This method helps achieve the complementary and generative contribution of ABM to analytical modeling, provides a formalized method for performing comparative epidemic studies and aids incremental modification and reproducibility of the agent model in order to achieve a realistically richer representation of the modeled phenomena. It also alleviates the lack of field data/lack of real geographical locations of the occurrence of particular cases by the creation of a benchmark model used for validation.

## Acknowledgements

I would love to thank Virginia Ejiofor, Njideka Mbeledogu and Moses Onyesolu for their support and valuable comments.

## References

- [1] M. Wooldridge, P. Ciancarini, *Agent-oriented software engineering : the state of the art*, Agent-oriented Software Engineering, Springer, Berlin Heidelberg, 2001, pp. 1–28.
- [2] W.O. Kermack, A.G. McKendrick, Contributions to the mathematical theory of epidemics, Proc. R. Soc. A 115 (1927) 700–721.
- [3] J. Pan, C.C. Fung, An agent based model to simulate coordinated response to malware outbreak within an organization, Int. J. Inf. Comput. Secur. 5 (2) (2012) 115–131.
- [4] R.B. Connell, P. Dawson, A. Skvortsov, Comparison of an Agent-based Model of Disease Propagation with the Generalised SIR Epidemic Model, (2009) .



- [39] H. Bersini, Article tools, *J. Artif. Soc. Soc. Simul.* 15 (1) (2012) 9–38.
- [40] Y. Shoham, Agent-oriented programming, *Artif. Intell.* 60 (1993) 51–92.
- [41] R. Axelrod, J.M. Epstein, M.D. Cohen, D. John, C.T. Macarthur, Aligning simulation models: a case study and results, *Comput. Math. Organ. Theory* 1 (1996) 123–141.
- [42] F.K. Batista, A. Martín del Rey, A. Queiruga Dios, Malware propagation in Wireless Sensor Networks: global models vs individual-based models, *Adv. Distrib. Comput. Artic. Intell. J.* 6 (3) (2017) 5–15, doi:<http://dx.doi.org/10.14201/ADCAIJ201763515>.
- [43] S. Peng, G. Wang, S. Yu, Modeling the dynamics of worm propagation using two-dimensional cellular automata in smartphones, *J. Comput. Syst. Sci.* 79 (2013) 586–595.
- [44] M. Birkin, B. Wu, A Review of Microsimulation and Hybrid Agent-Based Approaches A, in: J. Heppenstall, A.T. Crooks, L.M. See, M. Batty (Eds.), *Agent-based models of geographical systems* Springer Science+Business Media B.V, 2011.
- [45] A. Martín del Rey, A. Hernández Encinas, J.D. Hernández Guillén, J. Martín Vaquero, A. Queiruga Dios, G. Rodríguez Sánchez, An individual-based model for malware propagation in wireless sensor networks 13th International Conference Distributed Computing and Artificial Intelligence, (2016) .
- [46] A. Martín del Rey, G.R. Sánchez, A CA model for mobile malware spreading based on bluetooth connections, *Adv. Intell. Syst. Comput.* 234 (2014) 619–629.
- [47] G.G. García, M.E.L. Ramirez, Modeling the spatio-temporal dynamics of worm propagation in smartphones based on cellular automata 2016 European Modelling Symposium, (2016) .
- [48] S. Yurong, J. Guo-ping, Modeling malware propagation in wireless sensor networks using cellular automata, 2008 International Conference on Neural Networks and Signal Processing, (2008) , pp. 623–627.
- [49] Z. Bakhshi, M.Z. Lighvan, R. Mostava, MP-CA: a malware propagation modeling methodology based on cellular automata, *Int. J. Comput. Netw. Commun. Secur.* 3 (2015).
- [50] A. Bose, K.G. Shin, Agent-based modeling of malware dynamics in heterogeneous environments, *Secur. Commun. Netw.* 6 (2013) 1576–1589.
- [51] S. Hosseini, M. Abdollahi Azgomi, A. Rahmani Torkaman, Agent-based simulation of the dynamics of malware propagation in scale-free networks, *Simulation* 92 (2016) 709–722.