Appl. Math. Inf. Sci. 6 No. 2S pp. 603S-609S (2012)

Applied Mathematics & Information Sciences
An International Journal

An Architecture Model of Distributed Simulation System Based on Quotient Space

D. Wu [1], **X. J. Ban** [1], and **F. Oquendo** [2]

[1] Department of Computer Science and Technology, University of Science and Technology Beijing, Beijing, China.
[2] Department of Computer Science and Engineering, University of South Brittany, Vannes, France
Corresponding author: Email: wudi312858@gmail.com

Received June 01, 2011; Revised Sep. 03, 2011; Accepted Sep. 13, 2011
Published online: 1 January 2012

Abstract: It is an important and difficult work to build a consistent and efficient architecture for large scale distributed simulation systems based on their strong and complex standard constraints. On the system design level, the information we have is more abstract, fuzzy and inaccurate. So a new architecture modeling method is needed. In this paper, we propose an architecture design model based on the quotient space theory. The quotient space theory is an important theoretical branch of granular computing and concentrates on the relationships and architecture of the different granules. In our model, system design information is integrated into different types of granules, and intends for the further optimal design. This model considers the simulation tasks, resources and services as a whole, and gives a two-step modeling process on system architecture design, including task-resource allocation model and resource-service allocation model. We illustrate our method with an information system example to show the modeling and solving process. The primary result shows that it has reduced the period of system design and increased system consistency and efficiency.

Keywords: Distributed simulation systems, quotient space, software architecture, high level architecture.

1 Introduction

The development and application of large scale distributed simulation systems is now one of the most significant answers to the modern complex system research [1] including military training, economy, biological systems, and industrial processes.

Commonly accepted standards for distributed simulation systems, such as Distributed Interactive Simulation (DIS)[2] or High Level Architecture (HLA) [3-5] mainly describe the infrastructure supporting the simulation (e.g., simulators, devices, communication services, etc.) by providing the communication services, but lack of high-level concepts describing the functional decomposition of the overall simulation task. The resources allocation as well as the simulation tasks scheduling, yet key issues towards efficient simulations, consequently remain hand crafted and error prone.

This paper, we propose an architecture design model for distributed simulation system by using the concept of granulation [6-13]. Our contribution to these issues is twofold. On the one hand, we propose an abstract model, which properly captures the relationships of simulation tasks, available resources, existing services and logical constraints. On the other hand, we propose a method that enables the computation of a near optimal simulation architecture with respect to the global execution time and the workload of the simulation network.

Our solution leverages the quotient space theory[14-16] to break down the complexity of this optimization problem. This architecture model results from the granulation process advocated the quotient space theory and captures the simulation tasks, resources and services at the architecture level. It enables a two-step allocation method for the optimal system architecture design. We illustrate our approach on an image processing process, including acquisition, transmission and processing.

The remainder of this paper is organized as follows. Section 2 states the problem for distributed simulation system design and analyzed the possible solution. Section 3 recalls the inherent characteristics of distributed simulation systems and explains the formalization of distributed simulation system using quotient space theory. Section 4 illustrates the use of our approaches to find a near optimal architecture of image processing system. We conclude the work in section 5.

2 Problem Statement

2.1. Distributed simulation system analysis.

Distributed simulation system is a collection of simulation equipment in distributed areas interacted through Local Area Network (LAN) or Wide Area Network (WAN) with consistent structure, standard, protocol and database. The system structure is mainly in a star-shape. Each satellite node is responsible for calculating some part of the larger simulation task by sharing and broadcasting objects of interest. All the interactions have to comply with the interface standard for calling the services provided by the central node.

In a highly distributed and dynamic environment, a proper balance among simulation tasks, resource allocation and service assignment, is critical to distributed simulation system performance. The simulation tasks need to be allocated into different resources, and the corresponding simulation resources require calling the services to accomplish the specified task. But at design time, based on their different properties, it is difficult to give a formal model on the relationship between simulation task, resource, and service.

A set of simulation tasks needs to be declared by the designer. The simulation tasks can be divided into different levels, such as system, architecture, or program. Here we consider the tasks on the architecture level. The optimal design objective is to minimize the internal and external coordination workload and minimize the overall task completion time at the same time.

For distributed simulation system, the resources can involve humans, hardware and other operational function software to coordinate their actions for the achievement of their common goal. The capabilities of the resources can be divided into three parts, intelligence, hardware and network. We will discuss about the resource model in detail next section.

In order to make sure the simulators work together in the larger system through the central server, a set of services need to be defined. To avoid the bottle neck appears on this point, the service calling need to be balanced by all the simulators.

2.2. Quotient Space Theory.

Quotient Space is an important branch in granular computing theory. It intends to understand and describe the reality from different perspectives and levels and has been applied in pattern recognition, artificial intelligence, etc.

In the quotient space theory, a problem (or problem space) is described by a triplet (X, f, T), in which X is its domain, f is its attributes, T is its structure. Assume R is an equivalence relation on X, $[X]$ is a quotient set under R. Regarding $[X]$ as a new domain, and we have a new problem space ($[X]$, $[f]$, $[T]$). The worlds with different granule size are represented by a set of quotient spaces. The representation is intended to describe the worlds with different granule-size easily and can be used for analyzing the hierarchical problem solving behavior expediently. This simplification process is similar with the concept of quotient set in abstract algebra.

3 Model of Distributed Simulation Systems

The architecture of distributed simulation systems consist of the simulation tasks, all necessary resources and the services for interactions in the central server. At the same time, some specified constraints also need to be considered. In this section, we introduce the basic granular model for distributed simulation system and the two-step method for design, including task-resource and resource-service allocation models.

3.1. Basic Model

3.1.1 Basic problem

In quotient space theory, granulation of an object A leads to a collection of granules of A_i, with a granule being a clump of points (objects) drawn together by indistinguishability, similarity, proximity or functionality.

For the distributed simulation system architecture, the simulation tasks, resource and service can be abstracted to three independent quotient spaces based on their properties. And regarding with time and constraints, a problem of simulation system design could be described as a quintuplet model as follows:

$$P = < A, R, S, T, C > \quad (1)$$

In which,

A - represents the task information granule, where each A_i is a group of simulation tasks, such as, management, control, information processing, etc.

R - represents the resource information granule, where each R_i is a group of simulation resources, such as, computers, sensors, commander, etc.

S - represents the service information granule, where each S_i is a group of services, such as, time management, system management, interaction management, etc.

T - represents the time step;

C - represents the constraint set, including,

C_a - task constraints, mainly refers to the relationship between simulation tasks;

C_r - resource constraints, mainly refers to the capability of simulation resource;

C_s - service constraints, mainly refers to the requirements of the simulation service;

C_t - time constraints, mainly refers to the tasks need to be finished on the time limit.

3.1.2 Resource evaluation model

We divided the resource of distributed simulation system into three types, including intelligence, simulation entity hardware configuration and the network resource quality. They separately represent the intelligent ability, information processing ability and information transformation ability of the simulation unit.

There is one intelligent ability index. The hardware configuration indexes mainly include the response time, velocity of information access and rate of node in use. The indexes of the network resource quality mainly contain the network bandwidth, time-lag and rate of channel in use. These indexes can be calculated respectively. Accordingly, we can choose the weight parameters based on experience as the resource composite evaluation standard. Therefore the node resource evaluation of the distributed simulation system can be described as:

$$R(n) = \alpha_0 I(n) + \alpha_1 t_R(n) + \alpha_2 s(n) + \alpha_3 r_N(n) + \alpha_4 b(n) + \alpha_5 t_D(n) + \alpha_6 r_C(n) \quad (2)$$

α_i, ($0 \le i \le 6$), is the weight parameter. It is given by the designers. The other parameters represent the system evaluation indexes. There is

$$\sum_{i=0}^{6} \alpha_i = 1$$

$I(n)$ - represents the intelligence ability of the resource. In fact, here we divided the intelligence into five levels, and 5 is the highest level for the simulators.

$t_R(n)$ - represents the hardware entity response time at node n. It means the operation time distance from the information request to the information feedback, it is a random parameter for a single specified task with normal distribution.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, -\infty < x < \infty \quad (3)$$

$s(n)$ - represents the velocity of information access at node n. It means the information operation time for database. It can be assumed as a linear function of the information access quantity with the same hardware and software.

$r_N(n)$ - represents the rate of node in use. It means the rate of the node execution time and system execution time. This parameter indicates the efficiency of the system parallel execution.

$b(n)$ - represents the network bandwidth of node n. generally it can be set as a static value.

$t_D(n)$ - represents the network transformation time-lag at node n. It means the time distance from the information transformation request to the transformation finish confirmation. It is a random parameter with normal distribution.

$r_C(n)$ - represents the rate of network channel in use at node n. It means the rate between the net work transformation time and the system execution time. This parameter indicates the efficiency of the network.

3.2. Task-Resource Allocation Model

3.2.1 Assignment model

The multiplication of the task information granule vector and resource granule vector is a task-resource allocation matrix. The optimal design process is to find the possible value of $A_iR_j(t)$ for the required objective.

$$A \times R = \begin{bmatrix} A_1R_1(t) & A_1R_2(t) & \cdots & A_1R_n(t) \\ A_2R_1(t) & A_2R_2(t) & \cdots & A_2R_n(t) \\ \vdots & \vdots & \ddots & \vdots \\ A_mR_1(t) & A_mR_2(t) & \cdots & A_mR_n(t) \end{bmatrix} \quad (4)$$

In the matrix, t is the time parameter. $A_iR_j(t)$ is the value for the task i for resource j. And there is:

$$A_iR_j(t) = \begin{cases} 0 & R_j \text{ is not executing } T_i \text{ at } t \\ 1 & R_j \text{ is executing } T_i \text{ at } t \end{cases} \quad (5)$$

3.2.2 Basic assumption

In order to simplify the possible calculation, we set some primary assumptions, which are based on the distributed simulation system characteristics.

Assumption 1: The task granule at the architecture level cannot be further decomposed.

Assumption 2: Each task must and only can be executed once.

Assumption 3: Each resource can take one or more than one tasks independently.

Assumption 4: Each task can take at most one resource.

3.2.3 Optimization object

For the distributed simulation system, the execution time is the primary evaluation on the simulation efficiency. So, the possible optimization object of this problem could be minimizing the whole system execution time. Here we need a value weight matrix. This matrix decides the basic time for every resource to finish every task. The matrix can be defined as: So we get the formula as:

$$V_{AR} = \begin{bmatrix} V_{11} & V_{12} & \cdots & V_{1n} \\ V_{21} & V_{22} & \cdots & V_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ V_{m1} & V_{m2} & \cdots & V_{mn} \end{bmatrix} \tag{6}$$

In which, V_{ij} represents the time for resource j to finish task i, and there is $V_{ij}>0$ and

$$V_{ij}=t_{ijend}-t_{ijstart} \tag{7}$$

In which, $t_{ijstart}$ and t_{ijend} represent the start and end time for resource j to finish task i in the system. And $t_{ijend} \geq t_{ijstart} \geq 0$. So the optimal object could be as:

$$f = \min(\sum_{j=1}^{n}\sum_{i=1}^{m} V_{ij} \times A_i R_j) \quad 1 \leq i \leq m, \quad 1 \leq j \leq n \tag{8}$$

However in most of the practical system design problem, it is more valuable to find the satisfactory solution than the actual optimal solution according to the system requirements. If we can define the boundaries of the object as in function (9), this problem can be transferred to a Constraint Satisfaction Problem (CSP). In some cases, the efficiency of the problem solving can be increased.

$$\sum_{j=1}^{n}\sum_{i=1}^{m} V_{ij} \times A_i R_j < T_{acp} \quad 1 \leq i \leq m, 1 \leq j \leq n \tag{9}$$

In which, T_{acp} represents the acceptable solution.

3.2.4 Constraints

In the process to find the optimal solution, the different kinds of constraints also need to be considered, mainly including the task constraint, resource constraint and time constraint.

In practical problems, a single task normally has the constraint for execution time. For example, the longest time to execute task A_1 is T_{1max}. This can be described as:

$$T_{1end} - T_{1start} \leq T_{1\max} \tag{10}$$
$$\text{in which，} T_{1end} = \max(t), T_{1start} = \min(t), \ \forall A_1 R_i(t) = 1$$

The task constraint refers to the logical relationship between different tasks. Such as task A_2 needs the result from task A_1 to proceed, so A_1 has to be finished before A_2. This $A_1 << A_2$ could be described as:

$$T_{1end} \leq T_{2start} \tag{11}$$
$$\text{in which，} T_{1end} = \max(t), \forall A_1 R_i(t) = 1$$
$$T_{2start} = \min(t), \ \forall A_2 R_i(t) = 1$$

Resource constraint refers to the limitation of the resource capability, such as resource R_1 cannot carry on the task A_1. This could be as:

$$\forall t > 0, \ A_1 R_1(t) \neq 1 \tag{12}$$

3.3. Resource-Service Allocation Model.

3.3.1 Assignment model

The resource need to call the service to finish the task. The services have been defined as the interface standard [4] in distributed simulation systems. The service can be grouped as the information granule. The multiplication of the resource granule vector and service granule vector is a resource-service allocation matrix. In order to get the optimal resolution for the system design, we need to solve the matrix.

$$R \times S = \begin{bmatrix} R_1S_1(t) & R_1S_2(t) & \cdots & R_1S_k(t) \\ R_2S_1(t) & R_2S_2(t) & \cdots & R_2S_k(t) \\ \vdots & \vdots & \ddots & \vdots \\ R_nS_1(t) & R_nS_2(t) & \cdots & R_nS_k(t) \end{bmatrix} \tag{13}$$

In this matrix, R_iS_j represents the value of quantity that resource R_i calling S_j . and $1 \leq i \leq n$, $1 \leq j \leq k$. This is a decision matrix which is related to the resource and service allocation granule. And there is:

$$R_i S_j(t) \in [0,1], \quad when \, 1 \leq i \leq n, \ 1 \leq j \leq k \tag{14}$$

3.3.2 Basic assumption

Here we also set up some basic assumptions.

Assumption 1: if the resource does not take any task, then it cannot use the services provided by RTI (Runtime Infrastructure, the HLA service platform).

We describe it as

$$if \ \sum_{i=1}^{m} A_i R_j(t) = 0, \ then \ \sum_{l=1}^{k} R_j S_l = 0 \tag{15}$$

Assumption 2: $0 \leq R_jS_k(t) \leq 1$, this means the value of the resource calling for service is between 0 and 1.

3.3.3 Optimization object

The workloads of the network and the central node are important for the system execution efficiency. In order to make the data load of the satellite nodes in the star-like network to achieve basic dynamic balance, we can take the minimum mean square deviation as one of the optimization objects, as:

$$\min \sum_{i=1}^{n} [R_i S_j(t) - (\sum_{i=1}^{n} R_i S_j(t))/n]^2 \quad (16)$$

This is an optimization object of the interactive data get balance for each satellite node to service S_j. For the central node which has m classes of services, there are m different functions.

Generally speaking, in the processing of the central node, the same class of services run in serial process, different classes of services execute parallel process. Therefore, if the satellite-node requests focus on one service class in the central node, the processing efficiency will decrease inevitably. In order to make the system execution satisfied with the high real-time requirement, the service class requests from the satellite-node need to be balanced. However, for some particular satellite nodes, the service classes cannot achieve perfect balance. Different classes need different weight. Thus, the optimization object for the service load can be described as:

$$\min \sum_{i=1}^{k} [\beta_{ij} (R_i S_j(t) - (\sum_{j=1}^{k} R_i S_j(t))/k)]^2 \quad (17)$$

In this function, β_{ij} represents different weights with the same simulation node to different service classes in the central node.

3.2.4 Constraints

The constraints in Resource-Service assignment contain the constraint for resource, service and time.

The resource constraint can be some specified requirement to service, such as all the resources which join the system execution have to call the service granule S_p ($1 \le p \le$ k), this could be described as:

$$\forall i \in \{i \mid 1 \le i \le n\},\ \exists \sum_{t=1}^{T_{end}} R_i S_p(t) > 0 \quad (18)$$

For the service constraint has to comply with the corresponding standard. Such as, for all resources the precondition to use service S_p is to declare from service S_q, this relationship can be described as:

$$if\ R_i S_p(t_0) > 0,\ then\ \sum_{t=1}^{t_0} R_i S_q(t) > 0 \quad (19)$$

The time constraint refers to the time logic between both resources and services, and even for the whole system. For example, we assume a system X has m tasks, so the system time can be as:

$$\min(X_{start}) \ge T_{start},\ \max(X_{end}) \le T_{end} \quad (20)$$

4 Application on an Information System Design

With the help of the architecture model of distributed simulation system, we could make the system sound and efficient. The process has been defined in the model. Here we try to build a new simulation system which is for an image process simulation system, to show the method of the model.

We assume that the new system includes three simulation tasks, and two available resources and three service groups. The design is to shorten the system execution time with the efficient use of the simulation resource and services. The granules of the system architecture model could be described as follows:

[A] = {A_1(Image acquisition), A_2(Image transmission), A_3(Image processing)}

[R] = {R_1(Computer group 1), R_2(Computer group 2)}

[S] = {S_1(Management service), S_2(Time service), S_3(Data exchange service)}

Let the maximum time for each tasks to be T_1=5h, T_2=8h, T_3=10h. And assume that all the weights for different services are equal, as:

$$\beta_{11} = \beta_{12} = \beta_{13} = \beta_{21} = \beta_{22} = \beta_{23} = 1 \quad (21)$$

And the constraint is simple, task T_2 can just start after getting the result from task T_1. Based on the model described in section 2, we get the model of this problem, as follows.

Task-Resource matrix:

$$A \times R = \begin{bmatrix} A_1R_1(t) & A_1R_2(t) \\ A_2R_1(t) & A_2R_2(t) \\ A_3R_1(t) & A_3R_2(t) \end{bmatrix} \quad (22)$$

Resource-Service matrix:

$$R \times S = \begin{bmatrix} R_1S_1(t) & R_1S_2(t) & R_1S_3(t) \\ R_2S_1(t) & R_2S_2(t) & R_2S_3(t) \end{bmatrix} \quad (23)$$

In order to achieve the optimal design, we first need to find the feasible solution for Task-Resource matrix. By simple calculation, we can get the optimal time is 13 hours and there are basically 8 different feasible solutions. We can randomly pick one as follows:

$$AR(1) = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} \quad AR(2) = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad AR(3) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (24)$$

Based on this result, we further go in to solve the matrix of Resource-Service. This is a typical non-linear multi-objective problem. We use the Generic Algorithm to solve it. One of the acceptable results is as follows. It gives the allocation of the resources for the services from the server.

$$RS(1)=\begin{bmatrix} 0.683 & 0.237 & 0.08 \\ 0 & 0 & 0 \end{bmatrix}$$
$$RS(2)=\begin{bmatrix} 0 & 0 & 0 \\ 0.127 & 0.285 & 0.588 \end{bmatrix}$$
$$RS(3)=\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad (25)$$

The architecture result can be further developed and used into the program level of the system development. This architecture design model is in formal description, and could be solved by the existing mathematic methods. It could increase the efficiency for the system design process.

5 Conclusions

Facing the design problem of the large scale distributed simulation systems, to build a consistent and efficient architecture model with the simulation tasks, resources and services is always a crucial question. In this paper we briefly presented a formal architecture model of distributed simulation system using quotient space theory. Granulation process helps the designers to divide the problem hierarchically and solve it at different levels. Quotient space is an important granular computing theory which intends to describe the world with different granule size.

Regarding the system inherent qualities, this model considers all the simulation tasks, resources and services at the architecture level, and gives a two-step allocation method for the system design. This method has been applied to the system design on one of the Chinese 973 project which consists of more than fifteen participants during the system execution. Based on the development experiments, it reduced the period on system design and improved both the system consistency and efficiency.

Acknowledgements

We are grateful for the software resources provided by the European project ArchWare which aims at to provide a new approach for dynamic architecture evolution. And also, we thank Mr. Richard McClatchey for introducing us in Asia-Link project, discussing the connection between our subjects and use the libraries together.

This work wasSupported by National 863 Program(No. 2009AA062801)， National Nature Science Foundation of P. R. China (No.60973063) Beijing Natural Science Foundation of P.R. China (No. 4092028)，the Fundamental Research Funds for the Central Universities(FRF-TP-09-016B), Chinese Post-doctoral Science Funds (2010048199) and supported by the new century personnel plan for the Ministry of Education(NCET-10-0221).

References

[1] Hamilton J. A. Nash D. A. , Udo W. Pooch, *Distributed Simulation*, CRC Press (1997).

[2] IEEE 1278-1993, *IEEE Standard for Distributed Interactive Simulation – Application protocols*.

[3] IEEE Standard 1516-2000, *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)* - Framework and Rules.

[4] IEEE Standard 1516-2000, *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)* - Interface Specification.

[5] IEEE Standard 1516-2000, *IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA)* - Object Model Template.

[6] Miao D. Q, Wang G. Y, Liu Q, Lin Z. Y, Yao Y. Y, *Granular Computing: Past, Present, and Future* Science Press, (2007).

[7] Zadeh L A: Fuzzy sets and information granularity, *Advances in Fuzzy Set Theory and Applications*, North Holland, New York, 3-18, (1979).

[8] Zadeh L A. Towards a theory of fuzzy information granulation and fuzzy logic, *Fuzzy Sets and Systems*, vol.90, no.2, 111-127, (1997).

[9] Yao Y. Y., Granular Computing: basic issues and possible solutions. *In Proc. of fifth Joint Conference on Information Sciences*, Vol.1, Atlantic City, New Jersey, USA, 186-189 (2000).

[10] Yao Y.Y., and Li X., Comparison of rough-set and interval models for uncertain reasoning, *Fundamental Informatics*, 27: 289-298. (1996).

[11] Yao Y.Y. and Zhong N, *Granular Computing Using Information Table*, in T.Y. Lin, Y.Y Yao, and L. A. Zadeh (editors) Data Miming, Rough Sets and Granular Computing, Physica-Verlag, 102-124. (2000).

[12] Pawlak Z, *Rough Sets: Theoretical Aspects of Reasoning about Data*, Dordrecht: Kluwer Academic Publishers, (1991).

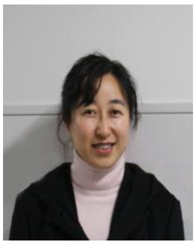
[13] Pawlak Z, Rough logic, *Bulletin of the Polish Academy of Science*, 35(5-6): 253-259 (1987).

[14] Zhang B, Zhang L, *Theory and Application of Problem Solving (2nd edition)*. Tsinghua University Press, (2007).

[15] Zhang Y.P, Zhang L, Wu T, The Representation of

Different Granular Worlds: A Quotient Space. *Chinese Journal of Computers*. (2004) Vol.27(3),:328-333.
[16] Zhang L, Zhang B, Granular computing based on quotient space model. *Journal of Software*, (2003), 14(4): 770-776.
[17] Wu Di, Chen Jie, and Oquendo Flavio, Optimal Design Model of Distributed Simulation System based on Quotient Space, J*ournal of Central South University (science and technology)* (2009), Vol. 40 Suppl. 1: 268-272.
[18] Chen Jie, Wu Di, and Zhang Juan, The Distributed Simulation System Hierarchical Design Model based on Quotient Space Granular Computation, *ACTA AUTOMAT-ICA SINICA*, (2010), Vol. 36(7): 923-930.

Di Wu received her BS degree from the Department of Automation, Beijing Institute of Technology in 2004. And she received Ph. D. on Pattern Recognition and Intelligent System at the School of Automation, Beijing Institute of Technology in 2010. She is currently a Post-doc researcher with the department of Computer Science in the University of Science and Technology Beijing, China. She has already published over 15 papers for international journals and conferences. Her research interest covers large scale distributed simulation system, optimal system design, and formal system development theory. (Corresponding author of this paper.)



Xiao-Juan Ban received her MS degree in computer science application from University of Science and Technology Beijing in 1997. And she received her Ph.D in Automation in 2003. She is currently a Professor with Department of Computer Science and Technology in University of Science and Technology Beijing China. She has published over 50 papers in international peer reviewed journals and conferences. Her research interest covers artificial intelligence, artificial life, and computer animation.

Flavio Oquendo is a professor and doctoral supervisor at the department of Software, the University of South Brittany. His researches areas are software architectures, formal methods, model-driven engineering and application.