Inheritance Growth: a Biology Inspired Method to Build Structures in P2P

Panchalee Sukjit, Herwig Unger

Abstract—IT infrastructures are becoming more and more difficult. Therefore, in the first industrial IT systems, the P2P paradigm has replaced the traditional client server and methods of self-organization are gaining more and more importance. From the past it is known that especially regular structures like grids may significantly improve the system behavior and performance. This contribution introduces a new algorithm based on a biologic analogue, which may provide the growth of several regular structures on top of anarchic grown P2P- or social network structures.

Keywords—P2P, Pattern generation, Grid, Social network, Inheritance, Reproduction

I. INTRODUCTION

BIOLOGICAL growth processes have fascinating properties for the observer. The reason for this is not only the variety of patterns which can appear, but also that the appearances of any creature allows it to survive in a given environment in an optimal manner, or at least better than its competitors. The respective properties are subject to an instant adaptation in an evolutionary process and may be inherited or recombined by children during the reproduction process.

Computer scientists and mathematicians tried several times to copy and use the properties of biologic, well working mechanisms for their purpose. Genetic algorithms, neural networks, ant algorithms and, last but not least, cellular finite state machines are just a few examples for doing so. Stephen Wolfram, in particular, shows in his book [1] how complex patterns taken from different life forms in nature may appear just by very simple algorithms. The most critics on the approaches existing so far are that

- connections between any appearing pattern or forms of the whole built structure and the environment either does not exist or cannot be studied or are hard to derive and to describe (most structures are symmetric);
- patterns appear in a previously fixed grid / cell universe only and often take advantage of an explicitly defined coordinate system;
- all cells have the same behavior;
- any cell differentiation is hard to be achieved.

Panchalee Sukjit and Herwig Unger are with the Dpt. of Communication Networks in the Faculty of Mathematics and Computer Science, Fernuniversität in Hagen, Germany. Universitätsstrasse 27 – PRG 58084 Hagen, Germany (phone: +49(0)2331987-1153; fax: +49(0)2331987-353; email: <given-name.surname>@fernuni-hagen.de).

As a result, pattern generation can seldom be efficiently used in (anarchic grown, unstructured) P2P systems to improve the performance or any related parameters of the system and to re-organize them under multi-criteria aspects. Existing algorithms to built so called structured decentralized networks, like for example Chord, Tapestry or CAN [2] fix the kind of appearing patterns from the beginning and low performance in case of (slower or faster) unexpected and dynamic changes happen in the network (like joining or leaving nodes). In [3] and [4] the authors suggested algorithms to build rectangular grid patterns, since more or less regular grid structures may provide significant advantages to the solutions of a lot of problems like routing and search [6,7]. In [3], the general computational requirements to any structure building methods on top of a decentralized system have been derived, too:

1. The structures must be built fast and with non-complex algorithms.
2. It must be easy to repair in case that any changes in the network appear.
3. The algorithm is running locally on each peer and therefore only can use the information available on this peer and eventually on its neighborhood peers (since global information is not available).
4. The generated overhead shall be minimal and the achieved efficiency maximal.

These works shall now be used as the basis or the design of a more general and flexible structure building algorithms on top of a P2P- and other decentralized technical systems. Therefore, the current contribution is structured as follows. In *section II* our new approach is formally described in 2 steps: at first a biology inspired cell reproduction mechanism is explained and then it will be discussed how it is used for grid generation. After this, *section III* considers the performance of the derived new approach by a set of simulations, while finally *section IV* concludes the paper and gives an outlook on future works.

II. A NEW GRID BUILDING ALGORITHM

A. The cell reproduction method

In nature, cell replication is controlled by chromosomes [4]. We intend to apply a similar mechanism to our cells, too. A few assumptions are made, to introduce the process in a

World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:3, No:11, 2009

simple manner and to avoid dealing with too many side-effects.

For that purpose it shall be agreed that:

1. We consider a tessellation of the Euclidian plane by square like cells q of side length 1.
2. Each square q may have exactly 8 neighbours (figure 4) and get 8 reproductions engines to generate new cells in those directions. Let us call the neighbours of $p \in \{1 ... 8\}$ a neighbour of q at port i.
3. At first, only the reproduction process is considered. At this moment it is not explained, how the neighbour connection is made, which let a grid appear. It will be later shown that it is possible to link all cells to a proper connected, rectangular grid.
4. It is assumed that all structures are finite and no infinite growth in any direction is allowed.
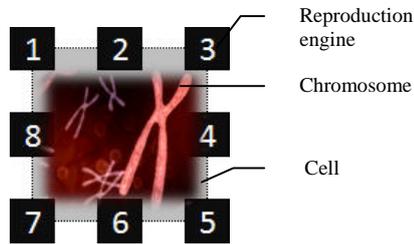

Fig. 1: The basic cell structure

We shall now describe a model of a cell growth in the tessellated plane. A cell c is a square in the plane whose genetic information for the reproduction process is represented by a genome Γ. The genome Γ consist of 8 chromosomes $\gamma^1 ... \gamma^8$. Here a chromosome is an $(m+1)$ tuple

$$\gamma = (x_1, s_1, ..., s_m)$$

where $\quad m \in \mathbf{N}_0, \quad x \in \{0, s_1, ..., s_m\}$

and $\quad s_i \in \{0, ..., m\}^8$

Let us call m the length and x the time to live $(TTL)^{1}$ of the chromosome γ. Note, that the 8 chromosomes of Γ do not need to have the same length. For simplicity we assume that all growth happens at discrete time intervals.
Consider a cell c born at time t.

Let $\quad \Gamma = (\gamma^1, ..., \gamma^8)$ be $c's$ genome

where $\quad \gamma^i = (x^i, s_1^i, ..., s_{m_i}^i)$

and $\quad s_j^i = \left(s_j^i(1), ..., s_j^i(8) \right)$

for $\quad i \in \{1, ..., 8\}, \quad j \in \{1, ..., m_i\}$,

At time $(t+1)$ the cell c gives birth to a new cell $\tilde{c}$ at port c, if and only if $x^i \neq 0$ and if there is no cell in the direction of port i. In this case the new cell $\tilde{c}$ at port i of c has the genome

^{1} Corresponding to the function of the "telomere", see http://en.wikipedia.org/wiki/Telomere.

$$\tilde{\Gamma} = \left(\tilde{\gamma}^1, ..., \tilde{\gamma}^8 \right)$$

when $\quad \tilde{\gamma}^i = \left(x^i - 1; \; s_1^i, \; ... \; s_{m_i}^i \right)$

and $\quad \tilde{\gamma}^j = \left(s_{x^i}^i(j), \; s_i^j ... \; s_{m_i}^j \right)$ $\qquad (*)$

*(Note: that by this conditions an infinite growth process become possible, in case the respective sequence parameters are set to -1 and N_0 is extended to **G**. Consequently in* $\ast$

$$s_{x^i}^i \quad \text{must be replaced by}$$

$$s^i_{\left(1 + |x^i| \bmod m_i \right)}$$

In a vector and matrix manner a genome is a tuple of vectors

$$\underline{\Gamma} = \left\{ \underline{x}, \; \underline{s}_1, \; ..., \; \underline{s}_m \right\}$$

where $\quad \underline{s}_1, \; ..., \; \underline{s}_m \in \mathbf{N}_0^{(8,8)}$

if $\quad x^i \neq o \left(i \in \{ 1, ..., 8 \} \right)$

The new genome of the newly generated cell c' can be presented by

$$\Gamma' = \left(\underline{y}, \underline{s}_1, \; ..., \; \underline{s}_m \right)$$

with

$$y^j = \begin{cases} s_{x_i}^{j,i}, & i \neq j \\ x^i - 1, & i = j \end{cases}.$$

The above described mechanism can defines

- a highly parallel growth process,
- whereby the special final form is determined by the sequences in the chromosomes.

With this setup we are now ready to discuss a grid building process within a P2P system in a more detailed manner.

B. Problems of grid generation by local processes

As already discussed in [2,3], the generation of completely populated, hole and prediction free grids may cause problems. These problems arise from wrong or not connected squares in the tessellated plane caused by an uncontrolled joining of new nodes in an only locally controlled building process (figure 2a).
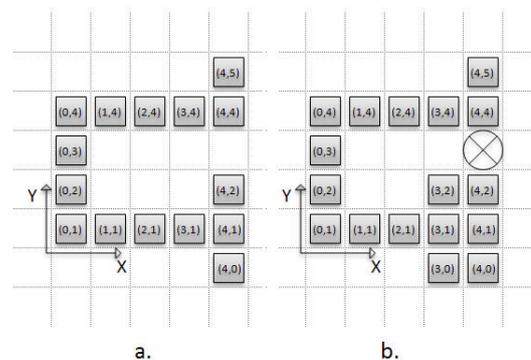

Fig. 2: Uncontrolled growth may result into holes and overlapping areas

World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:3, No:11, 2009

Another problem, which is implied by the existence of those holes, is that a node cannot see the situation behind gaps that come up with holes. This situation is shown in figure 2b at the position denoted with $\otimes$: Assume, that Node (4,2) accepts $\otimes$ to be its new neighbour. Since every node only knows about its direct neighbours, the node at position (4,4) does not know about the existence of $\otimes$. Node (4,4) still assumes that position (4,3) is free and therefore is ready to accept a new node at this position. That would lead to the contradictory situation that two nodes could reside at the same position (4,3). The rules on which the growth process is based must ensure that holes or contradiction situations cannot occur. Furthermore, it is of course also necessary to fix holes that arose from peer-failures within the grid.

C. Grid generation by inheritance growth

To describe the growing process we assume to have an initial cell system of 4 cells and for simplicity and later discussion we assume that they are centered in the origin of this coordinate-system, i.e. it has the coordinates (0, 0). To keep things easy we focus on the first quadrant. The discussed algorithm and its rules are symmetrically and therefore work analogous in the other three quadrants. Figure 3 shows the graphic representation of the growth rules coded in the chromosomes. In the cell of the considered 1st quadrant, only the chromosomes 2, 3 and 4 are active. A reproduction with chromosome γ^3 will result in a new cell, which obtains also active copies of chromosome 2 and 4, e.g.

$$s_j^3 = \left(0,-1,-1,-1,\ 0,\ 0,\ 0,\ 0\right), j \in \left\{1...m_3\right\}$$

(see figure 3b). The -1 denotes that we intend to have an infinite grid growth as long as new cells can grow/be added. Easier than this corner growth process are reproductions using port 2 and 4 for a horizontal or vertical growth only (figure 3 c and d). The respective chromosomes are simple, since only the already active chromosome must be inherited:

$$s_j^2 = \left(0,-1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 0\right), j \in \left\{1...m_2\right\}$$

$$s_j^4 = \left(0,\ 0,\ 0,-1,\ 0,\ 0,\ 0,\ 0\right), j \in \left\{1...m_4\right\}$$

a) Initial 4 nodes with the grow direction marks

b) Corner growth by adding new 3 nodes

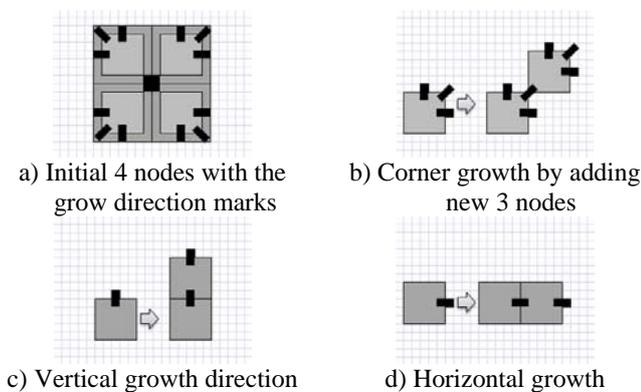c) Vertical growth direction

d) Horizontal growth

Fig. 3: The graphic representation of the growth rules

After the replication process, each new cell (locally) sends a message to its existing neighbors in its 1-neighborhood, to ensure the correct, rectangular grid linking. In a similar manner as in [3] can be proven that all necessary links will be so established in the right manner.
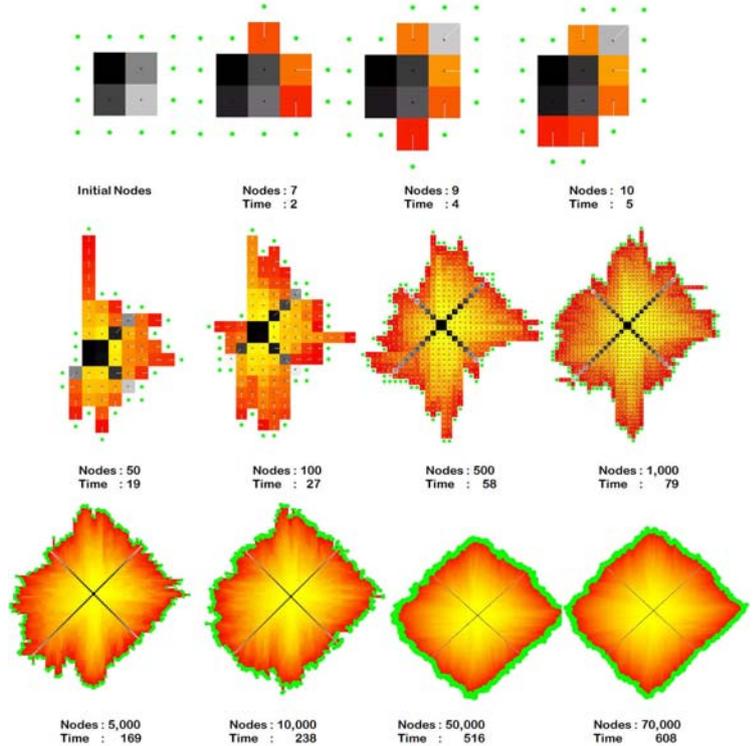

Fig. 4: Visualization of the grid growth in a simulation

Figure 4 shows the growth process. The application of the chromosome 3 for growth result in gray $\rightarrow$ black cells, the application of chromosome 2 and 4 in yellow $\rightarrow$ orange cells(and the respective symmetric chromosomes form the other quadrants), whereby darker cells are the younger (later included) ones. The green dots indicate the peripheral cells which are principally able to grow in the next simulation step.

In subsection A it was required that a hole-free and (with respect to the grid-like links between the cells) a contradiction-free structure must be created by our method. As figure 5 shows, every node (except the 4 root nodes) has precisely one ancestor or father who generates it. The investigation of this father son-relation obtained by our growth rules is shown in figure 6. It is easy to be seen and to be proven that always such tree like structures would appear by our methods and our initial genome. Since in such a manner all necessary links for the grid may be correctly established by knowing the current 1-neighborhood only, we obtain finally a hole- and contradiction-free grid structure. As shown in our former publications on grid building [3], the tree-like father-son relation also allows establishing the protocols, which ensure the respective fault tolerance in case of an unexpected leave of some nodes.

World Academy of Science, Engineering and Technology
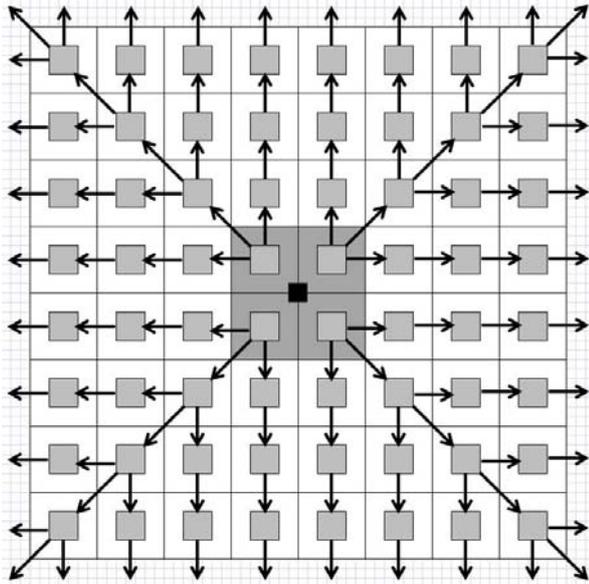International Journal of Mathematical and Computational Sciences
Vol:3, No:11, 2009

Fig. 5: Father-son relation between generated nodes

Last but not least, the infinite, almost uncontrollable growth or the convergence to a square is not always the wanted structure. As figure 6 motivates, the grid like structures shall cover a given area, like for instance a circle. Since the cell generation sequence can be directly derived from the figure (although the moment of a given cell inclusion/birth is not known in the respective random process) the contents of the chromosome sequences s^i of the genome Γ can be easily determined for _any_ needed border appearance of the grid. This is definitely the big advantage of our method, since in such a manner the construction process of the grid remains a local one.
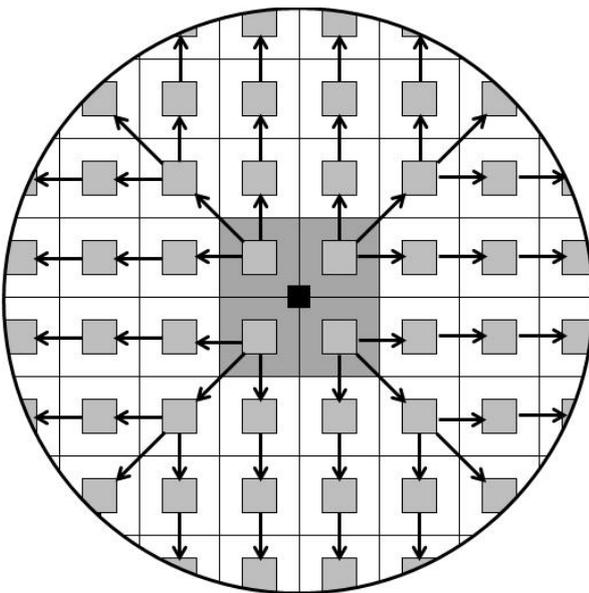

Fig. 6: Grid with a circle-like outer shape

III. PERFORMANCE EVALUATION

A. Simulation Setup

To observe, how this algorithm performs under real network conditions, P2PNetSim [8] – a distributed network simulator was utilized. P2PNetSim allows large scale network simulations and analysis on cluster computers (up to 2 million peers can be simulated on up to 256 computers). The behaviour of all nodes can be implemented in Java and then be distributed over the nodes of the simulated network. At simulation start-up the peers are interconnected small-world-like in order to simulate the typical physical structure of computers interconnected in the Internet. On top of this structure an overlay-network is built using the grid-algorithm. In order to explore the behaviour of our grid generation algorithm, search times for new nodes are not considered, i.e. a fixed constant time (6 steps) is assumed for that. Thus we avoid any interference of the duration of the search (fast in the beginning, slower when almost all nodes of the P2P community are already assembled in the grid) with the grid building algorithm. This is more important, since the parallelity of the growth process is rapidly increased with the number of nodes included in the grid.

B. Performance Evaluation

Based on the described methods the network growth process has been considered in several series of experiments. It was important to us to show its proper and efficient work with a relatively high number of nodes, i.e. 50,000-100,000.
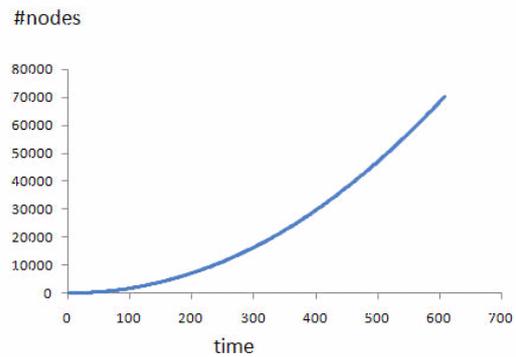

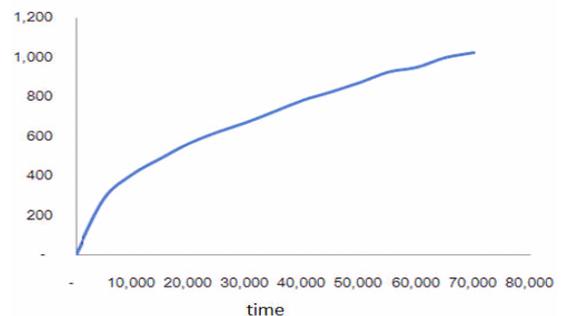Fig. 7: Number of nodes in the grid depending on time

Fig. 8: Number of nodes which can add new peers depending on the number of already included nodes

World Academy of Science, Engineering and Technology
International Journal of Mathematical and Computational Sciences
Vol:3, No:11, 2009

Figure 7 and 8 show the main results, which were obtained as a mean value from approx. 100 simulation runs.

It is clearly to be seen in figure 7 that the numbers of nodes grows approximately quadratic with time, i.e. the algorithm works pretty fast. The growth rate mostly depends on the number of peripheral nodes, but a growth rate of 25% is introduced. This means that per simulation step only approximately ¼ of the growable nodes are adding new cells, Note that the periphery does not growth in a linear manner with the number of nodes already included in the grid. In detail, the periphery grows only with the square root of the number of included cells/peers in the grid, which was confirmed in the simulation as shown in figure 8.

Finally must be mentioned that the growth rate may decrease significantly, if finite structures are generated. Any parts which reached their final size cannot add any more nodes and therefore reduce the number of peripheral cells which can reproduce themself.

IV.   CONCLUSION AND OUTLOOK

In the present contribution new methods for the generation of structures on top of anarchic grown P2P system (mostly small world structures) have been discussed. Based on the cell reproduction known form biology, a chromosome controlled reproduction process has been introduced. Using this approach, grid structures with any outer shape maybe generated; so far the generation of infinite, hole- and contradiction free grid structures has been discussed. Differing from other structure building methods, no predefined cell universe or coordination system is needed and all steps work in a local manner, i.e. only use the information of the considered call and/or send information from this cell to any direct neighbour one. The fast growth speed is another advantage of our new proposed method. This article only presents a basic biology inspired structure generation mechanism. Some simplifications where applied to it, which leave the following questions unanswered:

1. how chromosome information may be derived from a parameter description of the outer shape curve.
2. how a recombination of chromosome information or mutations may appear, since so far the genome is only copied in the reproduction process.
3. what happens if in the direction of port i is already a cell or 2 cells touch each other in the growth process.
4. how different types of cells may be reproduced depending on the chromosome and built much more manifold structures (other cell types, other than grid like connections between the cells).

These questions are the subject of the current research work of our team and result are expected to be published until the end of 2010.

REFERENCES

[1] Wolfram, S.: "A New Kind of Science", Wolfram Media, Champaign, IL, 2002.
[2] Lua, E. K; Crowcroft, J; Pias, M.; Sharma, R.; Lim, S.: "A Survey and Coparism of Peer-to-Peer Overlay Network Schemes" – IEEE Communications Survey and Tutorial 7, 2(2005), pp. 72-93, 2005
[3] Berg, D.; Coltzau, H.; Sukjit, P.; Unger,H.; Nicolaysen, J.: "Passive RFID-tag processing using a P2P architecture." Proc. of the 3rd Malaysian Software Engineering Conference, 2007.
[4] Berg, D., Sukjit, P., Unger, H.: "Grid Generation in decentralized Systems", Proceedings Intl. Workshop on Nonlinear Dynamics and Synchronization 2009 (INDS), pp. 95-99. Klagenfurt, Austria, 2009
[5] Hyde, D.: "Introduction to Genetic Principles", McGraw-Hill Publ. Company, 2008
[6] Mello, A.; Copello Ost, L.; Gehm Moraes, O.; Laert, N.; Calazans, V.: "Evaluation of Routing Algorithms on Mesh Based NoCs", Faculty of Informatics, Pontifícia Universidade Católica do Rio Grande do Sul, Brasil, Technical Report Series, No. 040, May 2004. Download from http://www.inf.pucrs.br/tr/tr040.pdf on July 20th, 2009
[7] Kleinberg, J.: "The small-world phenomenon: An algorithmic perspective", Proc. 32nd ACM Symposium on Theory of Computing, 2000
[8] Coltzau, H.: "Specification and Implementation of a Simulation Environment for Large P2P-Systems", Diploma, University Of Rostock, 2006

Panchalee Sukjit received an MBA from Ramkhamheng University in Bangkok. After 7 year in a managing position with Ajinomoto/ABL Thailand she became a PhD student at FernUniversity in Hagen in Germany in 2006 and works now in the area of decentralized distributed systems. Her special research interests are in algorithms for pattern generation and self-organization as well as computer simulations.

Prof. Dr.-Ing. habil. Herwig Unger got his PhD with a work on Petri Net transformation in 1991 from the Technical University of Ilmenau and his habilitation with a work on large distributed systems from the University of Rostock in 2000. Since 2006 he is a full professor at the FernUniversitaet in Hagen and the head of the Chair of Communication Networks. His research interests are in self-organization, adaptive and learning systems, Internet algorithms and simulation.