Publishing House
AKAPIT

COMPUTER METHODS IN MATERIALS SCIENCE
Informatyka w Technologii Materiałów

Vol. 9, 2009, No. 2

IMPLEMENTATION OF CELLULAR AUTOMATA FRAMEWORK DEDICATED TO DIGITAL MATERIAL REPRESENTATION

PATRYK SPYTKOWSKI, TOMASZ KLIMEK, ŁUKASZ RAUCH, ŁUKASZ MADEJ

Akademia Górniczo – Hutnicza, al. Mickiewicza 30, 30-059 Krakow, Poland
Corresponding Autor: lrauch@agh.edu.pl (Ł. Rauch)

Abstract

Design and implementation of programming framework dedicated to support development of the Cellular Automata (CA) algorithms is presented in the paper. The main objectives is to optimise the framework's architecture to obtain high runtime efficiency and flexibility. Various types of CA models starting with the simple one like the game of life finishing with the sophisticated models for simulation of physical phenomena can be implemented on the basis of this framework. The framework is combined with specific technology for visualization of obtained results that is based on OpenGL libraries and additionally is optimized regarding specific requirements of the CA method. This part of proposed computer system is described in the paper as well. The calculations of the CA based grain growth algorithm with different types of neighbourhood and space dimensions were performed to evaluate capabilities of the framework. Finally selected results are presented and discussed in the paper.

Key words: digital material representation, cellular automata, framework, visualization

1. INTRODUCTION

Development of reliable rheological models for metal forming simulations has been in the field of interest of scientists for a long time. A number of deterministic models, based either on closed form equations describing flow stress dependence on external variables [6] or on differential equations describing evolution of internal variables [5,12], have been published. These programs allow to describe in details the material behaviour under deformation and contribute to the general knowledge about the microstuctural phenomena that occur during metal processing and exploitation, e.g. hardening, recovery, recrystallisation, fatigues, fracture or crash. They are commonly used in industrial simulations of majority of metal forming processes and give satisfactorily accurate results. Nevertheless because they describe material as a continuum, they can only provide a general information about microstructural evolution i.e. average grain size distribution at the cross section.

Recently fast development of modern steel grades that are characterized by highly elevated material properties can be observed. These elevated properties are the results of very sophisticated microstructures with combination of e.g. large grains, small grains, inclusions, precipitations, nanoparticles, different phases, Luders bands, Portevin-le-Chatelier bands, shear bands etc. Interaction at the micro scale level between these microstructural features and the surrounding material under loading conditions directly results in mentioned properties at the macro scale level.

Problems with realistic description of phenomena occurring in materials at lower scales are the main limitation of commonly used continuum based method as far as mentioned above complex micro-

ISSN 1641-8581

structures are involved. One way to solve some of these limitations is to use multi scale models, which often use various numerical methods, e.g. Finite Element (FE), Boundary Element (BE), Extended Finite Element (XFE), Cellular Automata (CA), Cellular Automata Finite Element (CAFE), Molecular Dynamics (MD), connected together in one complex solution [10]. However, these methods still suffer from a lack of common numerical representation of material.

That is why to support experimental development of these new steel graders modern modelling approaches that take all microstructure and its features explicitly into account is required. Therefore, one of the solution is material models based on the Digital Material Representation (DMR).

2. STATE-OF-THE-ART

The concept of Digital Material Representation was recently proposed and is dynamically evolving [2,4,9,11,13,14,15]. The main objective of the DMR is creation of the digital representation of microstructure with its features (i.e. grains, grain orientations, inclusions, cracks, different phases etc.) represented explicitly. This approach allows to attach external numerical modules, which are able to simulate various microstructural phenomena like dynamic recrystallization, micro shear bands or fracture.

One of the most advanced software developed in [2] is called DIGIMAT. This software gives the possibility to create digital material samples and to test their properties before and after processing. In the case when obtained digital structure is not in a good agreement with the experimental data, a series of optimization algorithms is applied. Final structure is then the basis for the finite element mesh generation and subsequent FE simulations of the processing conditions. Similar approach is proposed in [4]. Both model are used to simulate microstructure development in large plastic deformations as well as during heat treatment (i.e. static recrystallization). Contrary to that another interesting work [1] is mainly focused on the elastic deformations. In [1] image based modelling of particle reinforced metal matrix composites and porous sintered steels is considered. The main advantage of this work is that digital microstructures are obtained directly from the microscope images prior to deformation. In 3D cases the digital microstructure is created based on the reconstructed 2D slices obtained using a destructive method.

Generation of material microstructure with specific properties is one of the most important algorithmic parts of systems based on the DMR. As mentioned such DMR is further used in numerical simulations of processing or simulation of behaviour under exploitation conditions. The more accurate the digital representation is the more accurate results can be obtained. Most commonly used method is Voronoi tessellation, however other method like Cellular Automata, Monte Carlo, Sphere Growth or Image Processing can also be found. Wide review of methods used for this purpose can be found in another Authors' work [3].

The main aim of this work is to develop not only a series of algorithms to generate initial microstructures but also to provide a general tool (CA Framework) for fast implementation of discrete material models that take into account different phenomena occurring during deformation i.e. dynamic and static recrystallization, phase transformation, shear band development etc. Due to the complexity of generated microstructures particular attention have to be put on efficiency in order to reduce the computational time.

The main assumptions regarding the CA Framework and its functionality are presented and discussed in the paper. This section is followed by description of visualization engine able to support presentation of large scale cubes. Examples of generated microstructures for one phase materials as well as microstructures with different shapes and sizes will be shown as well. Finally obtained digital microstructures will be prepared to be used as input data for the FE simulations to investigate e.g. influence of microstructural features on final material properties.

3. CA FRAMEWORK

3.1. Objectives and problems

As mentioned in the state-of-the-art section there is a lack of the universal programming framework able to support creation of a CA algorithms. All existing solutions have several disadvantages. Firstly, they are dedicated to specific problems like traffic simulation [7] or event forecasting [8], and are inaccessible for external users that are interested in implementation of their own algorithms. Therefore, the design and implementation of such open source framework is the first aim of this work. Secondly, available CA frameworks are characterized by low runtime efficiency. That is due to a large number of cells e.g. CA space with dimension 200 in 3D re-

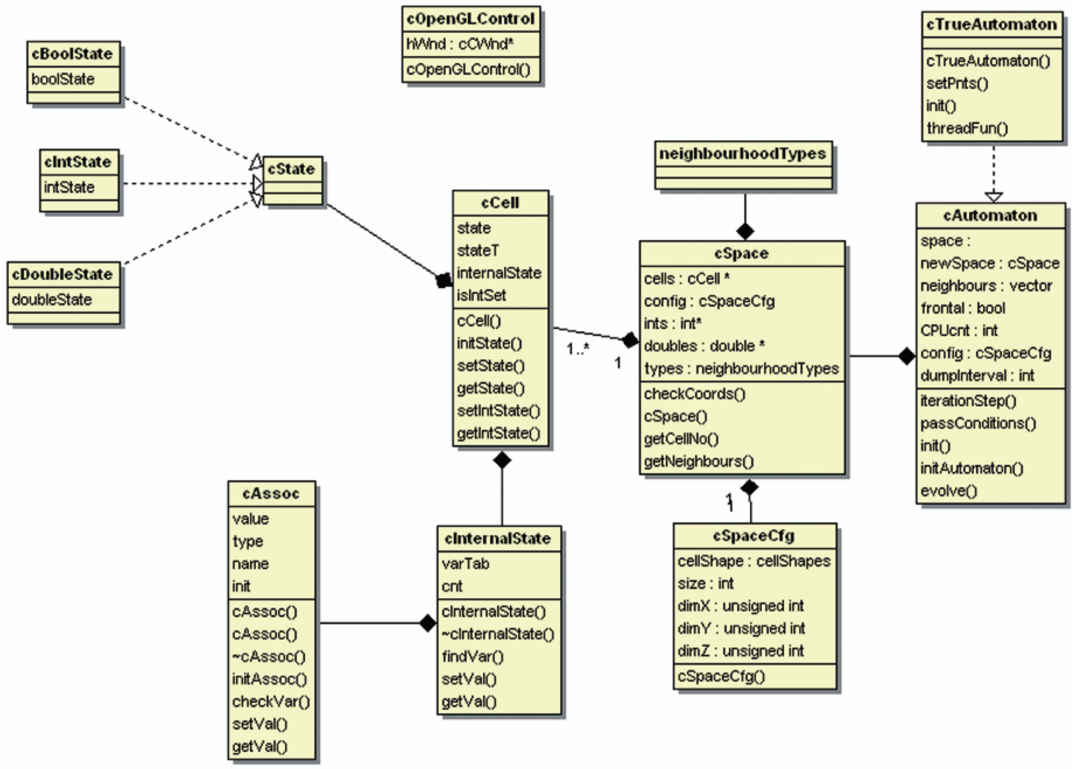

***Fig. 1.** Primary class diagram of the basic functionality of the proposed CA framework.*

quires 8E+6 cells and in case of 300 it is already 27E+6. Thus, simulation of huge number of cells is time, CPU and memory consuming. When a single cell consists only one variable of integer type (4 baits) the mentioned spaces already require 32MB and 108MB of memory respectively. Obviously, more sophisticated models require more variables, that represent various microstructure data/features and are used in the CA rules. In that case even more than gigabytes of RAM memory are required. These advanced CA models and algorithms force implementation of optimized codes regarding parallelization, efficient usage of cache memory and finally employment of specialized data structures. The solution proposed in this work satisfies all these needs and allow to perform parallel computations only in the CPU cache what significantly speed up the calculation process. In conventional approach with the RAM memory even the parallelization based on the OpenMP approach still results in deterioration of the final performance. This is caused by the loss of time during communication between multi-core CPU and shared RAM memory, which use the same data bus. In the modified approach the crucial factor influencing the efficiency of the CA framework was proper design of the code and division of the information sizes. The use of data divided into small pieces required by CPU cache allowed to improve the efficiency even by 6-7 times.

3.2. Classes

The developed architecture of the Cellular Automata framework consists of the two major classes: cSpace, cAutomaton and several additional components responsible mainly for data storing and configuration purposes (figure 1). The first mentioned class is used to store all the data related to CA space like dimensions or cells' pointers. It also provides a simple interface that is used by a particular automaton. Currently, the most important design and implementation issues are to achieve maximum efficiency and minimum memory usage. Early versions of the CA framework possessed class describing each single cell (cCell), however first attempts pointed out very low efficiency of memory usage by the operating system. To prevent a memory leaks this class was replaced with arrays, which store specific data of the CA cells. This solution also influences CPU usage and final efficiency.

The main objective of the cAutomaton class is to facilitate creation of individually designed CA algorithms. Therefore, the cAutomaton is responsible for abstract representation of automata and aggregates information about neighbours, configuration parameters, calculations data and two cellular spaces. One of this spaces is modified during the calculation of the transition rules, while the second one remains unchanged and provide data from the previous time step. The most crucial part during creation of a new CA algorithm is focused on implementation of three main cAutomaton methods providing the following functionality:

- configuration of the space as well as description of actions that take place at the beginning of automaton life cycle e.g. shape of cells, dimensionality, initial values of cells.
- implementation of a transition rules and boundary conditions using the cInternalStates as a class that is dedicated to gather internal variables e.g. dislocation density.
- description of neighbourhood of particular cell e.g. Neumann, Moore, hexagonal, pentagonal or randomized.

Additionally, the cAutomaton class also supports dumping of the CA cell states to the external file during each cycle of calculation. This action provides a possibility to restore the complete state of the CA space at each step of calculations. This functionality is used during the studying of evolution of particular phenomena. All functions mentioned above have to be implemented through the user defined class, which inherits from the cAutomaton to complete the functionality of a new CA algorithm. However, this step can be simplified and shortened by using the cTrueAutomaton class, which is dedicated to communication with other elements of application environment like visualization.

4. VISUALIZATION

The visualization of the results obtained from the CA framework is presented by the window application implemented on the basis of the OpenGL library. The primary version of the visualization algorithm assumed presentation of each cell in the form of six quads built from vertices, which were combined into cubes. However, this version of data presentation was highly inefficient, especially for spaces with dimension higher than two hundred cells. Therefore, the algorithm of visualization was optimized as follows:

- vertices shared between different cells are stored in one array of pointers (common sides of neighbouring cells are displayed only ones),
- only selected cells, which are visible for current scene have to be displayed in each cycle of the CA.

Examples of obtained 3D microstructures with different number of grains are presented in figure 2.

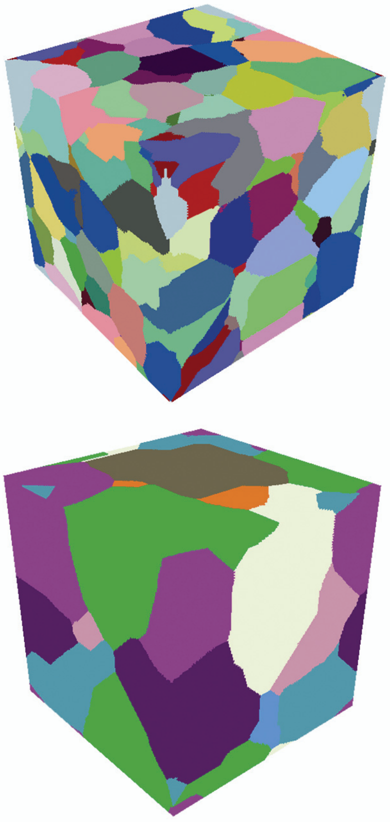

Fig. 2. *Screenshots from visualization module.*

With this optimized approach the problem of huge number of rendered vertices was partially solved and final view of the CA space of n-dimension includes about $48n^2$ vertices (table 1).

Table 1. *Vertices in comparison to the dimension and FPS ratio.*

Dimension	Vertices	FPS1	FPS2	FPS3
50	120 000	33	60	60
100	480 000	14	60	60
200	1 920 000	4	55	60

Last three columns in table 1 called FPS1, FPS2, FPS3 are related to frames per second ratios calculated using three following graphic devices respectively: GeForce4 MX440 with AGP8X 64MB, GeForce 7950 GT 512MB, GeForce 9600 GT 512MB.

The number of vertices increases exponentially influencing displayed frames per second and final quality of visualization. Therefore, the visualization module still requires improvements to present larger number of vertices during interaction with the user in real time mode. Low visualization capabilities can be currently caused by low efficiency of data transfer from RAM memory to the graphic device (each vertex is transferred separately). To avoid these encumbrances the combined array of vertices can be send directly to the GPU. Such functionality is satisfied by the Vertex Buffer

Object (VBO) tool, which is intended to enhance the performance of the OpenGL by application of vertex array as well as display lists (OpenGL tools), facilitating their implementations. The VBO allows vertex array data to be stored in high-performance graphics memory, which can reduce the number of function calls and redundant usage of the shared vertices, promoting efficient data transfer. The proper implementation of the VBO should results in multiplied FPS and fluent visualization of the 3D CA dim 400 space. That can help to obtain high run times even on regular graphical cards.

Speedup ratios calculated for 1, 2 and 4 CPU's for different dimensions of CA space are presented in figure 3. The obtained speedup is 0.409 for 50^3 space, 0.468 for 100^3 space and 0.55 for 200^3 space.

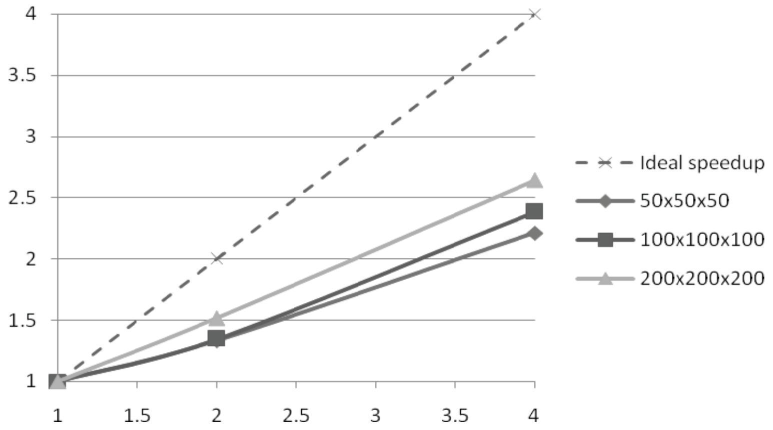

Fig. 3. Speedup plot related to three sizes of the CA space.

5. RESULTS

The proposed CA framework was tested using grain growth algorithm of the initial microstructure characterized by hexagonal neighborhood and constant number of grains inside CA space. The calculations were performed using three following machines to check the influence of the CPU cores number on finally efficiency:

Table 2. Results obtained for 100 grains in the CA space.

CORE NMBR	CELLS NMBR	CPU TICKS	TIME (s)	TIME (s) / CPU CYCLE	CPU TICKS / CELL	EFFICIENCY (SPEEDUP / CPU)
1	50^3	2 564 857	0.716531	0.053773	1.539854374	1.0
1	100^3	38 835 295	10.84923	0.42831	1.533155283	1.0
1	200^3	691 783 408	193.26	4.02625	1.801519292	1.0
2	50^3	1.29064E+9	0.536875	0.039291	755.6399153	0.667317
2	100^3	1.9711E+10	8.047887	0.318091	764.6904966	0.674042
2	200^3	3.0658E+11	127.395	2.514728	755.6766956	0.758507
4	50^3	778 865 883	0.323786	0.022117	425.6096448	0.553245
4	100^3	1.0958E+10	4.55564	0.184724	444.3520908	0.595374
4	200^3	1.7587E+11	73.113	1.453534	437.0592924	0.660826

(1) Intel Pentium CPU 2.66 GHz (Single Core), 2MB CPU Cache, 2GB RAM
(2) Intel Pentium Core 2 Duo e6600 CPU 2.66 GHz, 4MB CPU Cache, 4GB RAM
(3) Intel Pentium Core 2 Duo Quad q6600 CPU 2.66 GHz, 4MB CPU Cache, 4GB RAM

The obtained results are presented in table 2, where the following columns are related to number of CPUs, number of cells inside CA space, total time spent on calculations in seconds, the same time per CPU cycle, CPU ticks per cell and CPU ticks per cell multiplied by CPUs number.

6. CONCLUSIONS

The CA framework presented in the paper is characterized by high efficiency and satisfactory speedup ratios, which were obtained by performance of calculations using small portions of data allocated directly in the CPU cache memory. The designed framework proved to be useful in many practical applications e.g. simulation of initial grain growth. It was also successfully applied to simulate recrystallization phenomenon. However, advanced simulation models are still in the development phase. The optimization of efficiency of the proposed framework, which can be obtained by using specific properties of CPUs, data alignment inside physical memory or grouping common cells' properties in one data structure will be also performed during future work. Development plans assume generation of highly sophisticated multiphase microstructure of materials with randomized properties and inclusions. Such artificially created microstructure could be superimposed with finite element mesh and subjected to various numerical simulations. Finally, this approach will allow to obtain the most reliable representation of material in digital form, which can be analyzed further in many practical manufacturing processes.

ACKNOWLEDGEMENTS

The financial support of the Polish Ministry of Science and Higher Education, project no. R15 012 03, is acknowledged.

LITERATURE

1. Ayyar, A., Chawla, N., Microstructure-based modeling of crack growth in particle reinforced composites, Composites Science and Technology, 66, 2006, 1980-1994.
2. Bernacki, M., Chastel, Y., Digonnet, H., Resk, H., Coupez, T., Logé, R.E., Development of numerical tools for the multiscale modelling of recrystallisation in metals, based on a digital material framework, Comp. Meth. Mater. Sci., 7, 2007, 142-149.
3. Cybułka, G., Jamrozik, P., Wejrzanowski, T., Rauch, L., Madej, L., Digital representation of microstructure, Proc. CMS Conf., Krakow, 2007, 379-384.
4. Dawson, P.R., Miller, M.P., The digital material – an environment for collaborative material design, project poster. Available at: http://anisotropy.mae.cornell.edu/downloads/dplab/.
5. Estrin, Y., Dislocation Density related constitutive modelling, in: unified constitutive laws of plastic deformation, eds. Krausz, A.S., Krausz, K., Academic Press, 1996.
6. Grosman, F., Application of the flow stress function in programmes for computer simulation of plastic working processes, J. Mat. Proc. Technol., 64, 1997, 169-180.
7. Hatzi, O., Thomas, S., Dalakas, V., Nikolaidou, M., Anagnostopoulos, D., A cellular automata framework for studying expandable traffic flow models, Proc. of the 2007 summer computer simulation conference, 2007 (CD).
8. Karafyllidis, I., Thanailakis, A., A model for predicting forest fire spreading using cellular automata, Ecological Modelling, 99, 1997, 87-97.
9. Madej, L., Gawad, J., Hodgson, P.D., Pietrzyk, M., Contribution to digital representation of materials subjected to thermo-mechanical processing, Proc. MS&T, Detroit, 2007, 403-414.
10. Madej, L., Mrozek, A., Kuś, W., Burczyński, T., Pietrzyk, M., Concurrent and upscaling methods in multi scale modelling – case studies, Comp. Meth. Mater. Sci., 8, 2008, 1-10.
11. Melchior, M.A., Delannay, L., A texture discretization technique adapted to polycrystalline aggregates with nonuniform grain size, Comp. Mater. Sci., 37, 2006, 557-564.
12. Pietrzyk, M., Finite element based model of structure development in the hot rolling process, Steel Research, 61, 1990, 603-607.
13. Rauch, L., Madej, L., Pietrzyk, M., Hybrid system for model ling and optimization of production chain in metal forming, J. of Machine Eng., 8, 2008, 14-22.
14. Rauch, L., Madej, L., Deformation of the dual phase material on the basis of digital representation of microstructure, Steel Res. Int. 79, Special Edition Metal Forming Conf., 2, 2008, 247-254.
15. Romanova, V., Balokhonov, R., Makarov, P., Schmauder, S., Soppa, E., Simulation of elasto-plastic behaviour of an artificial 3D-structure under dynamic loading, Comput. Mater. Sci. 28, 2003, 518-528.

IMPLEMENTACJA FRAMEWORK'U DO AUTOMATÓW KOMÓRKOWYCH DEDYKOWANEGO CYFROWEJ REPREZENTACJI MATERIAŁÓW

Streszczenie

Niniejsza praca przedstawia projekt oraz implementację framework'u programistycznego dedykowanego do wspomagania rozwoju algorytmów opartych o technologie automatów komórkowych. Główny cel pracy skupia się na optymalizacji architektury framework'u tak, aby osiągnąć jednocześnie wysoką efektywność obliczeniową oraz elastyczność w możliwości projektowania nowych algorytmów. Obecnie różnego rodzaju algorytmy oparte o automaty komórkowe mogą być definiowane na bazie zaproponowanego framework'u, począwszy od prostej gry w życie, a skończywszy na skomplikowanych algorytmach zarodkowania i rozwoju mikrostruktur materiałowych. Framework połączony jest ze specjalnym modułem wizualizacji otrzymanych wyników zaimplementowanym za pomocą technologii OpenGL. Rezultaty uzyskane dzięki połączeniu obydwu modułów zostały również przedstawione i omówione w niniejszym artykule.

Submitted: October 28, 2008
Submitted in a revised form: November 25, 2008
Accepted: November 25, 2008