UDC 004.832

S. I. Shapovalova, Ph. D.,
O. O. Mazhara, Post-graduate

National Technical University of Ukraine
«Kyiv Polytechnic Institute», Kyiv

PRODUCTION SYSTEM SHELL FOR MOBILE DEVICES

The CLIPS shell extension was proposed to enable the creation of applied production systems which can be used on mobile devices. The innovation was substantiated in terms of saving money resources to purchase specialized equipment for using for the resource-intensive production systems. The following studies were conducted: market research tablet devices, the most widely represented in Ukraine; production systems shells that can be used on mobile devices; freely distributable production systems shells were considered, important characteristics were identified and compared. The basic classes of match algorithms were overviewed. The results of studies about Rete and Treat match algorithms advisability were described for the general case of the applied problems. The proposed modeling environment for production systems for mobile devices will reduce development time and increase system efficiency by choosing the optimal match algorithm for minimal memory usage.

Key words: *production systems; match algorithm; tablet computer.*

Introduction. One of the main approaches to solving intellectual tasks is based on models of knowledge representation. The largest segment of the knowledge-based systems is a production system (PS).

It is reasonable to use an existing productions systems tool for automating the solution of applied problems. Its compulsory components are empty knowledge base with specified format for records representation and inference engine. The production system developer`s task is filling the knowledge base with the information that is needed for inference in a particular subject area.

Existing production systems development tools are widely used in the software market for implementation of the technical and medical diagnostics, management and forecasting systems [1–3]. Production systems which are developed on the basis of such tools are used in the education, agriculture, financing activities, stock exchange [4–5].

Production systems operation requires significant computing resources. Therefore, it is considered a priority that such systems are installed on desktop computers. However, nowadays it is a common requirement for software to be compatible with different mobile devices such as smartphones or tablets. It

should be borne in mind that these devices may have different operating systems and a broad range of hardware implementation.

The amplification of the mobile devices contributed to the development of specialized software for employees of different areas, like guards and conductors of the railway transport, agronomists, sanctuary staff, and rescuers. Expert systems in agriculture, medicine, education became available to the public through the integration with the mobile devices [6].The development of such software puts rigorous requirements to memory usage and support of the interoperability between different platforms. This is why, the problem of choosing of the basic algorithms for production system inference, which will reduce memory usage, and developing software for this task is the objective of the day.

Inference engine of production systems based on the two components:

- match algorithm (for selecting productions to the conflict set);
- implementation of the inference strategy (conflict resolution).

The majority of modern development environments allow users to select from the main strategies of conflict resolution (depth, breadth, simplicity, complexity, novelty, LEX, MEA), and provide the implementation of auxiliary strategies for the effective inference (refraction, random selection, sorting by priority). At the same time, only one match algorithm is previously built in software tools. Moreover information about this algorithm is often not available for developer of the current production system. It is proved that the matching is the most demanding task in the inference process [7].Therefore, the choice of algorithm for implementation in the development environment of production systems has a significant impact on its effectiveness.

This study aims to create additional development tools of production systems that allow to select optimal solutions in terms of memory usage, with the use of match algorithm for the current problem, to conduct approbation of the chosen algorithm and to use it's implementation for the system which is developed.

Research Market Mobile. In order to identify software requirements for mobile devices Ukraine market was analyzed and characteristics of the represented products were identified.

The specificity of production systems is the first necessity of providing a convenient interactive interface for the user, and secondly the need for direct data entry. Therefore, tablets are a priority among mobile devices for the use of production systems. Using data from the most popular sites for the sale of mobile devices Ukrainian tablet market was analyzed on the basis of characteristics such as price, random access memory (RAM) and read only memory (ROM) [8], [9] . The results are presented in Table 1.

Table 1

Characteristics of the tablets on the Ukrainian market

Price ($)	RAM)	ROM	The market share
<85	19% — 1 GB 78% — <=0.5 GB	4 GB	8%
85 — 165	76 % — 1 GB, 21% — <=0.5 GB	8 GB	32%
165 — 290	74 % — 1 GB 23% — 2 GB	16 GB — 65% 8 GB — 27 %	34%
290 — 415	50 % — 1 GB 42% — 2 GB	16 GB — 50% 32 GB — 30 %	10%
415 — 585	41% — 2 GB 33 % — 1 GB 19% — <=0.5 GB	16 GB — 37% 32 GB — 36 % 64 GB — 25 %	7%
> 585	42 % — 1 GB 27% — 2 GB 26% — >=3 GB	16 GB — 18% 32 GB — 25 % 64 GB –33 % 126 GB — 21 %	9%

According to the data, a mid-range (165 — 415 $) dominated the tablet with 1-2 GB of RAM and 16 GB of ROM. Based on this analysis it can be concluded that the current amount of the internal memory is enough for the use of the modern production systems. At the same time, RAM resources of mobile devices are significantly inferior to desktop computers nowadays. The developers of operating systems for mobile devices have to create methods for efficient memory management. Better utilization of memory on the tablet is complicated by the fact that a physical increase the RAM for the existing devices impossible or financially impractical.

Currently, devices with Android operation system are the most common on the mobile market, far ahead of his closest rivals iOS and Windows. It is predicted that this trend will not change by 2017, although it is projected that market share of mobile devices that use the operation system iOS will increase (Market share held by smartphone operating systems worldwide in 2013 and 2017).

The specific feature of Android is the limit on memory available application to ensure the execution of multiple programs simultaneously. In Android, every application runs in a Linux Process. Each Linux Process has a Virtual Machine (Dalvik Virtual Machine) running inside it. There is a limit on the memory a process can demand. It is different for different devices. For most of the early versions of the operating system, this limitation was 16 Mb, for version 2.3-32 Mb, and from 4th version — the default is allocated 64 Mb. When some process demands a higher memory than limit it causes a forcibly termination with different types of errors due to lack of memory. The developer can change the size restrictions on the

memory usage for the program, but this decision is the worst practice and it is unacceptable to create cross-platform applications [10].

Android manages applications that are stored in memory automatically: in case there is a shortage in memory, the system starts to terminate applications and processes that have been inactive for some time, in reverse order, starting with one that has not been used for the longest time. This process is designed to simplify the operation with the memory for user who does not need to explicitly shutdown the application. So even correctly implemented production system for desktop computers will work perfectly and at the same time could prematurely shut down improperly on mobile devices in case of lack of memory.

The effectiveness of the applied production systems on mobile devices is primarily determined by the inference engine embedded in it. In turn, the effectiveness of the inference defined by match algorithm.

Match algorithms. Basic approach of modern algorithms for pattern matching is incremental matching [11]. The basis of this approach is keeping from cycle to cycle some information about the state of the system, known as a state support. In turn, incremental match algorithms are divided into 3 classes: lazy evaluation algorithms, binding space algorithms and eager evaluation algorithms. The latest is the most common in production system shells.

Lazy evaluation algorithms provide the integration of the match and conflict resolution, which leads to a number of limitations in the process of inference. A well-known example of such an algorithm is Leaps, that was proposed by Miranker [12] to improve the performance of expert systems.

Binding space algorithms aimed at facilitating a parallel implementation to increase performance on modern hardware. The most known from those algorithms are Matchbox [13], HAL [14], GridMach [15]. However, despite the considerable amount of research work in this direction, the suppliers of production systems tell nothing about their widespread use.

In view of the aforementioned, this work focused on the eager evaluation algorithms, whose effectiveness has been proven by their use in the modern production systems shells. The best known among these algorithms are Treat [12], Rete [16], Gator [17], Rete* [18] and their modifications.

Rete algorithm is, historically, the most used in the production systems shells as first developed. It is characterized by significant speed and has proven effective in many application problems [19]. At the same time, many researchers based on experimental studies, prefer Treat algorithm as one that requires less computational resources. Comparison of the Rete and Treat algorithms and guidelines for selecting one of them are presented in the following works [20].

Research shows that the efficiency of the match algorithm depends on how you present the information in the knowledge base. Therefore, it is impossible to make generalized conclusions for choosing algorithmic base of the production systems. However, the above recommendations allow a

high probability to choose an optimal algorithm in term of performance memory usage. It is important to create an environment that will make possible approbation of such a choice.

Rete and Treat match algorithms has been chosen for implementation on mobile devices. Treat algorithm was chosen due to less memory costs compared to Rete while it preserves considerable speed. An additional advantage of the chosen algorithm is that it was created for computers with parallel processing, which allows using it for modern multi-core mobile devices.

Production systems shells. Since the common production systems are developed by specialized production systems shells it is advisable to choose one of them for use on tablets.

Production systems shell is a software package that facilitates the construction of production systems by providing a scheme of knowledge representation and inference mechanism. The developers have to add knowledge related to the subject area; provide interconnection with existing software and information databases.

This study focused on the analysis of production systems shells that can be used on mobile devices. Table 2 presents data about such environment, including examples of their use.

Table 2

Production systems shells

N	PS shell		Application	
	Name	**Developer**	**Name**	**Purpose**
1	Exsys Corvid [21]	Exsys Inc	Jet Aircraft Diagnostics	Aircraft Diagnostics [22]
2	OPS [23]	Charles Forgy at Carnegie Mellon University	XCON	Determining the configuration of a computer system [24]
3	CLIPS [25]	NASA's Johnson Space Center, now maintained independently from NASA as public domain software	PI-in-a-Box	Aid for astronauts to perform science experiments in space [26]
4	Jess [27]	Sandia National Labs	Rice Plant Disease Diagnosis	Diagnosis of rice plants [28]
5	e2gLite [29]	eXpertise2GO	Expert system for diagnosing tropical infectious diseases	Diagnosing tropical infectious diseases [30]

These shells are widespread due to such characteristics as ease of use, possibility of integration with external systems, supports multiple paradigms of knowledge representation, cross-platform implementation and height performance. In this study the shells which are distributed under a free license were examined in more detail. These shells and their formal characteristics are presented in Table 3.

Table 3

Freely distributable production systems shells

Name	Knowledge representation	Cross-platform	Object-oriented approach	Procedural approach	Integration with external environments
CLIPS	Rules, objects	Compatible with most Windows and Linux operating systems, the latest versions of Android	Support	Support	Possible integration with C / C++, Java languages
Jess	Rules, objects	Compatible with most Windows and Linux operating systems, the latest versions of Android	Support	Support	Possible integration with Java language
eXpertise2GO	Rules, Decision tables	Compatible with most Windows and Linux operating systems, the latest versions of Android	Not support	Not support	Possible integration with HTML language

The shell eXpertise2GO is commonly used to create applications for the Android operation system, but it has several disadvantages: restrictions on the knowledge representation, focuses on web application, denied access to the source code. Jess is development environment for production systems, which was created on the basis of CLIPS for expert systems for developing with the use of Java programming language. However, this environment is distributed freely for non-commercial use, and requires an additional license for usage in commercial purposes.

CLIPS was chosen for further study among the following shells as best variant for applied tasks implementation. This is due to its features such as multiple programming paradigms support: the rule based, procedural, object-oriented; open source code; full supporting documentation; the possibility of cooperation with the developers, availability of the versions for different operating systems. An additional advantage for choosing the CLIPS environment was the possibility of its integration for IOS operating systems and MacOS. CLIPS can be ported to any system where C compiler compatible with ANSI is presented. CLIPS can be called as a subprogram and integrated with other programming languages such as C, C ++, Java, FORTRAN and Ada. In addition, the shell can be easily extended by using several well-defined documented protocols.

Extended production systems shell. In the CLIPS, development environment only Rete match algorithm was implemented. In this research we extended the functionality of this environment with Treat algorithm.

CLIPS for Android which called CLIPS4Android are the Android libraries which are intended to be used by other Android applications. This provides an additional advantage because there are no limits on memory

allocation in the native part of the program in Android OS. Modifications have been made so that the user can compile a library with one of the matching algorithm or with both. At the design stage of production system it is advisable to use the latest version.

Both algorithms were formalized in same format to minimize dependence from specific programing language. The both algorithms have compilation and execution part. Building data flow network from rule base is meant by compilation. Such data flow network represents data dependencies between rule conditions. At run-time facts from working memory enter network and are preceded depended on networks node type. Rete algorithm is formally described in terms of first-order logic in research [31]. Based on this research execution of Treat algorithm was formalized as follow.

Define in the token form fact f which has been added/remove from working memory (WM) $< +, f >$. For intra-elements test (conditions in one single pattern) the transition rule $?m \xrightarrow{a} ms!$ from one Treat network to another is formalized as follows:

$$< +, f > ? \xrightarrow{lm_j \leftarrow lm_j \cup \{f\}} \{+, f\}! \;\; ifq(f),$$

$$< -, f > ? \xrightarrow{lm_j \leftarrow lm_j - \{f\}} \{-, f\}! \; ifq(f),$$

$$< + | -, f > ? \rightarrow 0! \;\; if \neg q(f),$$

where *lm* is subset of fact that have been matched already, $q(f)$ the conjunction of all tests that have to be done for matching constant element.

For testing the inter-element conditions, conditions which involve more than one single pattern transition rule is formalized as follows:

$$\forall f \in lm_j < -, f > ? \longrightarrow \{< -, f >\};$$

$$\forall f \in lm_j < +, f > ? \xrightarrow{tm_j \leftarrow tm_j \cup \{f\} \wedge \sigma \leftarrow \sigma \cup \sigma' | \exists R \in PM \wedge \exists p \in p^+(R) | \sigma' p = f} 0!,$$

$$ifq(f) \wedge \neg r_j(tm_j);$$

$$\forall f \in lm < +, f > ? \xrightarrow{tm_j \leftarrow tm_j \cup \{f\} \wedge \sigma \leftarrow \sigma \cup \sigma' | \exists R \in PM \wedge \exists p \in p^+(R) | \sigma' p = f} \{< +, tm_j >\}!,$$

$$ifq(f) \wedge r_j(tm_j);$$

$$\forall f \in lm < + | -, f > ? \rightarrow 0!, \; if \neg q(f).$$

For conflict set CS:

$$< -, f > ? \xleftarrow{cm \leftarrow cm - \{f\}} \{< -, f >\},$$

$$< +, tm_j > ? \xrightarrow{cm \leftarrow cm \cup tm_j;} \{< +, tm_j >\}!$$

where R — production, σ — a substitution, p — pattern, which is defined as term, p^+ — positive pattern, r_j — the conjunction of all inter-element tests of rule $l_j \in PM$ (Production memory), which have to be

done for bind variables correctly, *tm* — subset of facts from WM with already bound variables, *cm* — conflict set memory — subset of facts from WM which match conflict set rules.

Given such formalization it is easier to compare algorithms regardless of the specific implementation.

Since Treat algorithm was developed on the basis of Rete, they have a similar mechanism to handle productions. What's more Rete and Treat algorithm have the same input and output format. So same data structure can be used for both algorithms. This allows the user correctly compare the performance of these algorithms for solving applied problem as information for them is presented in the same format.

It has been proved that Treat algorithm needs from 15 to 45% less memory than Rete [12]. In mobile devices such improvement can allow using of expert systems for low-end tablets. While for High-Ram devices user can make choose between memory usage and performance depending on the task. For such cases, the recommendations based on the properties of knowledge base are useful in selecting an algorithm. It was proved that an efficiency of production system depends on the properties of the knowledge base. The conducted experiments and previous researches allow forming the basic guidelines to select match algorithm which are presented in table 4.

Table 4

Recommendations for choosing match algorithm

Characteristics of the problem	Optimality criterion	Rete	Treat
A large number of conditional elements in the antecedent (> 6)	Number of comparisons with conditional elements	+	–
average number of conditional elements in antecedent does not exceed 6	Number of comparisons with conditional elements	–	+
High temporal redundancy	Performance	+	–
A large number of negative conditional elements	Performance	–	+
	Memory usage	–	+
Conditional elements are added dynamically	Performance	–	+
	Memory usage	–	+
Minimum number of complex variables in antecedent	Performance	+	–

Thus, the proposed development environment for production systems for mobile devices will reduce development time and increase system efficiency by choosing the optimal match algorithm for minimal memory usage.

Conclusions. Modern production systems for solving tasks of technical and medical diagnosis, management, forecasting, and education designed for use on desktop computers. This causes difficulties when transferring such systems on mobile platforms due to the specificity of their memory management. However, this problem can be solved by the use of optimal algorithms of production systems inference for specific tasks. This study focused on match algorithm for production systems for mobile devices.

As results of the study, the following conclusions can be made.

- Mobile devices most suitable for the application of production systems were explored. The expediency of using the tablet with Android OS was determined.
- The pattern match algorithms were investigated as a main component, which affects the efficiency of production systems. The following algorithms were chosen for the design of production systems on mobile devices: Rete, Treat.
- Production systems shells that can be used on mobile devices were examined. The CLIPS usage was justified.
- Highlight disadvantage CLIPS: implementation of only one match algorithm. Due to the open source and free license shell extension was created. Treat algorithm was proposed for that.
- Recommendations for choosing an appropriate match algorithm for applications were presented.

References:

1. Khan J. Prospector. (Scribd Inc). — Access mode: http://ru.scribd.com/-doc/44131016/Prospector-Expert-System.
2. Moore R. Expert System Applications in Chemistry. In G2: Chemical Process Control / R. Moore. — Cambridge : ACS Symposium Series, 1989. — P. 169–179.
3. Kamau Gichah H. Rule-based Process Support for Enterprise Information Portal / H. Kamau Gichah. — Hamburg-Harburg : Technische Universitдt Hamburg-Harburg, 2003 — 63 p.
4. D'Alessio R. An Expert System for Financial Accounting Teaching / R. D'Alessio // System Dynamics Society. — 2002. — P. 1–18.
5. PUFF. AI Systems in Clinical Practice. — Access mode: http://www.openclinical.org/aisp_puff.html.
6. Movafegh Ghadirli H. An Adaptive and intelligent Tutor by Expert Systems for Mobile Device / H. Movafegh Ghadirli, M Rastgarpour // International Journal of Managing Public Sector Information and Communication Technologies. — 2012. — № 3 (1). — P. 21-28.
7. McDermott, J. The efficiency of certain / J. McDermott, A. Newell, J. Moore // Pattern-Directed Inference Systems — 1978. — P. 155–176.
8. Hotline. (2014). Retrieved from Hotline. — Access mode: http://hotline.ua/-computer/planshety.
9. Price.ua. (2014). Price. — Access mode: http://price.ua/catc6399t1.html.
10. Dubroy P. Memory management for Android Apps. — Access mode: https://www.youtube.com/watch?v=_CruQY55HOk.
11. Bergmann G. Graph and model transformations / G. Bergmann, G. Varry // Incremental Pattern Matching in the VIATRA Model Transformation System. — Budapest : Department of Measurement and Information Systems, 2008. — P. 25–32.
12. Miranker D. Treat: A better match algorithm for ai production / D. Miranker // National Conference on Artificial Intelligence. — 1987.
13. Perlin M. Match box: Fine-grained parallelism at the match level / M. Perlin // Tools for Artificial Intelligence: IEEE International Workshop. Fairfax: IEEE Computer Society Press. — 1989. — P. 428–434.

14. Lee P. HAL: a faster match algorithm / P. Lee, M. Cheng // IEEE Transactions on Knowledge and Data Engeneering — 2002 — P. 1047–1058.
15. Tan J. A basis for integrating production systems with relational databases / J. Tan, M. Maheshwari, J. Srivastava // Tools for Artificial Intelligence: IEEE International Workshop. — Herndon : IEEE Computer Society Press, 1990. — P. 400–407.
16. Nayak P. Comparison of rete and treat production matchers for soar / P. Nayak // Artificial Intelligence: Seventh National Conference. — Cambridge : The MIT Press, 1988 — P. 693–698.
17. Hanson E. Gator: An optimized discrimination network for active database rule condition testing / E. Hanson. — Florida : CIS Departement, 1993.
18. Wright I. The execution kernel of rc++: Rete*, a faster rete with treat as special case / I. Wright // International Journal of Intelligent Games and Simulation. — 2003. — P. 36–48.
19. Laerhoven K. Comparison of the CLIPS and JESS expert system shells / K. Laerhoven // Lancaster : School of Computing and Communications. — 1999.
20. Wang Y. A performance comparison of the rete and treat algorithms for testing database rule conditions / Y. Wang, E. Hanson // Washington: IEEE Computer Society Press. — 1992.
21. Exsys Corvid Expert System Development Tool. Retrieved from Exsys. Knowlenge Automation Expert Systen Technology. — Access mode: http://www.exsys.com/exsyscorvid.html.
22. Exsys Case Study. Jet Aircraft Diagnostics. Retrieved from Exsys. Knowledge Automation Expert System Technology. — Access mode: http://www.exsys.com/-winkPDFs/CessnaSelfServiceDiagnostics.pdf.
23. Forgy C. The 0PS83 Report / C. Forgy // Department of Computer Science Carnegie-Mellon Univerrity. — Pittsburgh, 1984.
24. Barker V. Expert systems for configuration at Digital: XCON and beyond / V. Barker, D. O'Connor, J. Bachant, E. Soloway // Communications of the ACM. — 1989. — P. 298–318.
25. Riley G. CLIPS. A Tool for Building Expert Systems. Retrieved, from CLIPS Documentation. — Access mode: http://clipsrules.sourceforge.net.
26. PI-in-a-Box : A Knowledge-Based System for Space Science Experimentation / Frainier, Richard, Groleau, Nicolas, Hazelton, Lyman, Colombano, Silvano, Compton, Michael, Statler, Irving, Szolovits, Peter, Young, Laurence // IAAI '93 Proceedings of the The Fifth Conference on Innovative Applications of Artificial Intelligence. — 1993. — P. 34–49.
27. JessRules. Retrieved, from Jess the Rule Engine for the JavaTM Platform. — Access mode: http://www.jessrules.com.
28. Robindro K. JESS Based Expert System Architecture For Diagnosis Of Rice Plant Diseases:Design And Prototype Development / K. Robindro, S. Sarma // Intelligent Systems Modelling & Simulation. — 2013. — P. 674–676.
29. Expertise2Go Retrieved from Expertise2Go Web-Enabled Expert Systems. — Access mode: http://expertise2go.com/webesie/e2gdoc.
30. Putral I. K. Fuzzy Expert System for Tropical Infectious Disease by Certainty Factor / I. K. Putral, P. M. Prihatini // TELKOMNIKA. — 2012. — P. 825–836.

В роботі запропоновано розширення програмної оболонки CLIPS для забезпечення можливості створення прикладних продукційних систем, використовуваних на мобільних пристроях. Обґрунтована інновація с то-

чки зору економії грошових ресурсів на придбання спеціалізованих пристроїв для використання ресурсоємних продукційних систем. Проведено наступні дослідження: ринку планшетних пристроїв, найширше представлених в Україні; оболонок продукційних систем, які можуть застосовуватися на мобільних пристроях; більш детально, з виділенням значимих характеристик, розглянуто вільно поширювані обгортки продукційних систем. Описано основні класи алгоритмів співставлення зі зразком. Представлені результати досліджень щодо доцільності застосування Rete та Treat для загальних випадків в прикладних задач.

Ключові слова: *продукційні системи, співставлення зі зразком, портативні пристрої.*

Отримано: 18.03.2015

УДК 519.6:539.3

М. Р. Петрик, д-р фіз.-мат. наук, професор

Тернопільський національний технічний університет
імені Івана Пулюя, м. Тернопіль

МОДЕЛЮВАННЯ ТА ІДЕНТИФІКАЦІЯ ПАРАМЕТРІВ ЗАДАЧ ДИФУЗІЇ В НЕОДНОРІДНИХ НАНОМУЛЬТИКОМПОЗИТАХ ГРАДІЄНТНИМИ МЕТОДАМИ ТА ІНТЕГРАЛЬНИМИ ПЕРЕТВОРЕННЯМИ

Розроблена та обґрунтована на основі теорії оптимального керування станами багатокомпонентних систем, інтегральних перетворень Фур'є та Лапласа методика моделювання та функціональної ідентифікації параметрів дифузії для неоднорідних наномультикомпозитів, отримані аналітичні вирази градієнтів функціоналів нев'язок, компонентами яких є побудовані аналітичні розв'язки прямої і сряженої вихідної неоднорідної крайової задачі переносу та спряженої їй за Лагранжем задачі, виконано відновлення коефіцієнтів дифузії для різних поверхонь спостережень.

Ключові слова: *масоперенос, математична модель, функціональна ідентифікація, інтегральне перетворення, неоднорідні середовища, градієнт функціонала невязки.*

Вступ. Дослідження дифузійних процесів в багатошарових нанокомпозизитів і тонких наноплівок відкриває перспективи створення на базі матеріалів з відомими властивостями матеріалів та середовищ з новими властивостями (нові явища провідності, дифузійно-адсорбційні ефекти тощо), виникнення яких головно пов'язано зі структурними змінами середовищ при агрегатуванні наношарів з різними властивостями [1–4; 15]. В цій праці на прикладі задач дифузії в згущених (Fe/Dy) магнітних багатошарових наноплівках, утворених агрегацією наношарів з високопровідними і низькопровідними матеріалами (феромагнетики і