INTERNATIONAL JOURNAL

of

COMPUTERS, COMMUNICATIONS & CONTROL

With Emphasis on the Integration of Three Technologies

IJCCC

Year: 2010 Volume: V Number: 2 (June)

Agora University Editing House

CCC Publications

www.journal.univagora.ro

EDITORIAL BOARD

Editor-in-Chief

Associate Editor-in-Chief

Executive Editor

Managing Editor

Associate Executive Editor

ASSOCIATE EDITORS

Short Description of IJCCC

Title of journal: International Journal of Computers, Communications and Control
Acronym: IJCCC
International Standard Serial Number: ISSN 1841-9836, E-ISSN 1841-9844
Publisher: CCC Publications - Agora University
Starting year of IJCCC: 2006
Founders of IJCCC: I. Dzitac, F.G. Filip and M.J. Manolescu
Editorial Office:

R&D Agora Ltd. / S.C. Cercetare Dezvoltare Agora S.R.L.
Piaţa Tineretului 8, Oradea, jud. Bihor, Romania, Zip Code 410526
Tel./ Fax: +40 359101032

E-mail: ijccc@univagora.ro, rd.agora@univagora.ro, ccc.journal@gmail.com
Website: www.journal.univagora.ro

Number of issues/year: IJCCC has 4 issues/odd year (March, June, September, December) and 5 issues/even year (March, September, June, November, December). Every even year IJCCC will publish a supplementary issue with selected papers from the International Conference on Computers, Communications and Control.

Coverage:

- Beginning with Vol. 1 (2006), Supplementary issue: S, IJCCC is covered by Thomson Reuters - SCI Expanded and is indexed in ISI Web of Science.
- Beginning with Vol. 2 (2007), No.1, IJCCC is covered in EBSCO.
- Beginning with Vol. 3 (2008), No.1, IJCCC, is covered in SCOPUS.

Scope: IJCCC is directed to the international communities of scientific researchers in universities, research units and industry. IJCCC publishes original and recent scientific contributions in the following fields: Computing & Computational Mathematics; Information Technology & Communications; Computer-based Control.

Unique features distinguishing IJCCC: To differentiate from other similar journals, the editorial policy of IJCCC encourages especially the publishing of scientific papers that focus on the convergence of the 3 "C" (Computing, Communication, Control).

Policy: The articles submitted to IJCCC must be original and previously unpublished in other journals. The submissions will be revised independently by at least two reviewers and will be published only after completion of the editorial workflow.

Contents

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 2, pp. 144-147

Acad. Solomon Marcus at 85 Years (A Selective Bio-bibliography)

F. G. Filip (Editor in Chief of IJCCC),
I. Dzitac (Associate Editor in Chief of IJCCC)

Professor Solomon Marcus is one of the *seniors* of the Romanian science and culture, with a tremendously diverse and intensive activity, with numerous, fundamental, and many times pioneering contributions to mathematics, mathematical linguistics, formal language theory, semiotics, education, and to other areas, with an impressive internal and international impact and recognition, since several decades very actively involved in the scientific and cultural life – and still as active as ever at his age. That is why the short presentation below is by no means complete, it is only a way to pay our homage to him at his 85th birthday anniversary.

We are honored and we thank to professor Solomon Marcus for having the kindness to publish, in the first issue of this journal, an article occasioned by the Moisil centenary[1].

1 Biography and General Data

Born at March 1, 1925, in Bacău, Romania. Elementary school and high school in Bacău, classified the first at the school-leaving examination (“bacalaureat"), in 1944. Faculty of Science, Mathematics, University of Bucharest, 1945-1949. Assistant professor since 1950, lecturer since 1955, associate professor since 1964, and professor since 1966. Professor Emeritus since 1991, when he retired. All positions held in the Faculty of Mathematics, University of Bucharest. PhD in Mathematics in 1956 (with the thesis about “Monotonous functions of two variables"), State Doctor in Sciences in 1968, both at the University of Bucharest. Corresponding Member of Romanian Academy since April 1993, and Full Member of the Romanian Academy (Academician) since December 2001.

[1] Grigore C. Moisil: A Life Becoming a Myth, International Journal of Computers, Communications & Control, 1(1),73-79, 2006

Research and teaching in the fields of mathematical analysis, theoretical computer science, measure theory, general topology, linguistics, history and philosophy of mathematics, poetics, semiotics, applications of mathematics to natural and social sciences.

Has published over 50 books, as single author or in collaboration, and over 400 research articles, as well as several hundreds of articles on various cultural topics. Has edited over 40 collective volumes.

More than 250 invited lectures at various universities from Europe, Americas, and Asia.

In his papers and books has solved many open problems in mathematical analysis and has formulated a still larger number of problems and research topics, has introduced new ideas, notions and models, which were addressed and continued by a large number of researchers. Just an example: the contextual grammars he has introduced in 1967-68 are now called *Marcus contextual grammars* and were subject of many papers and PhD theses, as well as of two monographs.

Many students and collaborators, a really “contagious" professor and researcher.

Cited and followed by thousand of authors, from Romania and abroad.

His books were reviewed and commented by many personalities, among whom, we mention a few: Emanuel Vasiliu, Mike Holcombe, Sheng Yu, Yehosua Bar-Hillel, Maurice Gross, L. Nebesky, Michel Janot, I.I. Revzin, Ferenc Kiefer, H.P. Edmundson, Ju.A. Sreider, Jacob L. Mey, Carlo Tagliavini, Gr.C. Moisil, Matei Călinescu, Nicolae Manolescu, Sorin Stati, K. Sgallova, P. Sgall, Gabriela Melinescu, Jean-Marie Klinkenberg, Virgil Nemoianu, Adrian Marino, Miron Nicolescu, Alexandru Ivasiuc, Wladimir Krysinski, Mircea Mihăieş, Şerban Cioculescu, Eugen Simion, Constantin Negoiţă, Tudor Octavian, Dinu Flamând, Stelian Tănase, Alin Teodorescu, Mircea Scarlat, Smaranda Vultur, Cristian Calude, Gheorghe Păun, Tudorel Urian, Paul Cernat, Irina Mavrodin, Rosa del Conte, Jean-Pierre Descles, Umberto Eco, Marius Iosifescu, Roman Jakobson, Gheorghe Mihoc, Octav Onicescu, Cesare Segre, Gheorghe Tomozei, Iorgu Iordan, Constanţa Buzea, Cornel Ungureanu, Bogdan Suceavă, Basarab Nicolescu, Alexandru T. Balaban, Mihai Zamfir, and many others.

Member of the editorial board of over three dozens of professional journals.

Among the numerous honors received, we recall: Doctor Honoris Causa of the Universities of Bacău, Constanţa, and Craiova. Vice President of the International Association for Semiotic Studies (1989-1999). Honorary President of the Balkan Semiotic Society (since 2001). Member of the Executive Committee of the International Association for Semiotic Studies. Honorary Member of the International Association for Visual Semiotics. Honorary Member of the Toronto Semiotic Circle, of Como (Italy) Semiotic Circle and of the Hungarian Semiotic Society. Honorary member of the Romanian Mathematical Society. Permanent Invited Professor, Sao Paulo Catholic University. Member of the Romanian National Mathematical Committee. Member of the leading Council of the Romanian Linguistic Society. Member of the Romanian Union of Writers and Member (1990-1995) of the leading Council of this Union. Vice-president of the Romanian Semiotic Society. Member of the Romanian Committee for the Club of Rome. Honorary member of the Society Eratosthene, Switzerland.

He has also received many prizes, in Romania and abroad.

With various occasions, special issue of journals and collective volumes were devoted to him. Here are some of them: *The International Journal of Computer Mathematics* (Great Britain) and *Revue Roumaine de Mathematiques Pures et Appliquées* devoted some of their issues in 1985 to his 60th anniversary, while two collective volumes were dedicated to his 70th anniversary, in 1995, *Mathematical Aspects of Natural and Formal Languages*, World Scientific, Singapore, 1994, and *Mathematical Linguistics and Related Topics*, Ed. Academiei, Bucharest, 1995. In the year 2000, two volumes were dedicated to his 75th anniversary: *Finite vs Infinite. An Eternal Dilemma*, Springer, London, and *Recent Topics in Mathematical and Computational Linguistics*, Ed. Academiei, Bucharest. In the year 2005, volume 64, numbers 1-4 of *Fundamenta Informaticae* were published as a “Special issue in Honor of the 80th Birthday of Professor Solomon Marcus", with papers by 70 authors from 16 countries.

2 Books, Authored or Co-authored

1. *Lingvistica matematică. Modele matematice în lingvistică.* Ed. Didactică şi Pedagogică. Bucureşti, 1963.
2. *Gramatici şi automate finite*. Ed Academiei, Bucureşti, 1964.
3. *Analiza matematică.* Vol. I. Ed. Didactică şi Pedagogică, Bucureşti, 1st edition 1962, 2nd edition 1963, 3rd edition 1966, 4th edition 1971, 5th edition 1980 (in collab. with M. Nicolescu and N. Dinculeanu).
4. *Lingvistica matematică* (2nd edition, revised and completed with 4 new chapters). Ed. Didactică şi Pedagogică, Bucureşti, 1966.
5. *Introducere în lingvistica matematică.* Ed. Ştiinţifică, Bucureşti, 1966 (in collab. with E. Nicolau and S. Stati)
6. *Noţiuni de analiză matematică. Originea, evoluţia şi semnificaţia lor*. Ed. Ştiinţifică, Bucureşti, 1967.
7. *Limbaj, logică, filozofie*. Ed. Ştiinţifică, Bucureşti, 1968 (in collab. with Al. Boboc, Gh. Enescu, C. Popa, and S. Stati).
8. *Analiza matematică.* Vol. II. Ed. Didactică şi Pedagogică, Bucureşti, 1st edition 1966, 2nd edition 1971, 3rd edition 1980 (in collab. with M. Nicolescu and N. Dinculeanu).
9. *Introduction mathematique a la linguistique structurale*. Dunod, Paris, 1967.
10. *Algebraic Linguistics; Analytical Models*. Academic Press, New York, 1967.
11. *Poetica matematică*. Ed. Academiei, Bucureşti, 1970.
12. *Teoretiko-mnozestvennye modeli jazykov*. Ed. Nauka, Moscova, 1970 (translation of the first five chapters of book 10 and of the last chapter of book 9).
13. *Algebraicke modely jazyka*. Ed. Academia, Prague, 1969 (translation in Czech of book 4, and of a part of book 2).
14. *Introduzione alla linguistica matematica.* Casa editrice Riccardo Patron, Bologna, 1970 (revised and completed translation of book 5; in collab. with E. Nicolau and S. Stati).
15. *Mathematische Poetik*. Ed. Academiei, Bucureşti – Athenaum-Verlag, Frankfurt am Main, 1973 (revised and completed translation of book 11).
16. *Matematicka Poetika*. Ed. Nolit, Belgrad, 1974 (revised and completed Serbo-Croatian translation of book 11).
17. *Din gândirea matematică românească*. Ed. Ştiinţifică şi Enciclopedică, Bucureşti, 1975.
18. *Semiotica folclorului. Abordare lingvistico-matematică.* Ed. Academiei, Bucureşti, 1975 (co-author).
19. *Matematicka analyza ctena podruhe*. Ed. Academia, Prague, 1976 (revised and completed Czech translation of book 6).
20. *A nyelvi szepseq matematikaja*. Ed. Gondolat, Budapesta, 1977.
21. *Metode distribuţionale algebrice în lingvistică*. Ed. Academiei, Bucureşti, 1977 (coauthor).
22. *La semiotique formelle du folklore. Approche linguistico-mathematique*. Ed. Klincksieck, Paris – Ed. Academiei, Bucureşti, 1978 (revised and completed translation of book 18; co-author).
23. *Introduccion en la linguistica matematica.* Ed. Teide. Barcelona, 1978 (revised and completed Spanish translation of book 5).
24. *Semne despre semne*. Ed. Ştiinţifică şi Enciclopedică, Bucureşti, 1979.

25. *Contextual ambiguities in natural & artificial languages*. Vol. 1, *Communication and Cognition*, Ghent, Belgium, 1981 (revised and completed translation of a part of book 21).
26. *Snmeia gia ta snmeia*. Ed. Pneumatikos, Atena, 1981 (Greek translation of book 24).
27. *Metode matematice în problematica dezvoltării*. Ed. Academiei, Bucureşti, 1982 (co-author).
28. *Gândirea algoritmică*. Ed. Tehnică, Bucureşti, 1982.
29. *Semiotica matematică a artelor vizuale*. Ed. Ştiinţifică şi Enciclopedică, Bucureşti, 1982 (coordinator and co-author).
30. *Simion Stoilow*. Ed. Ştiinţifică şi Enciclopedică, Bucureşti, 1983 (in collab. with Cabiria Andreian Cazacu).
31. *Paradoxul*. Ed. Albatros, Bucureşti, 1984.
32. *Timpul*. Ed. Albatros, Bucureşti, 1985.
33. *Artă şi ştiinţă*. Ed. Eminescu, Bucureşti, 1986.
34. *Analiza matematică*. Vol. II, Univ. Bucureşti, 1986 (co-author).
35. *To Paradocso*. Ed. Pneumatikos, Atena, 1986 (Greek version of book 31).
36. *Şocul matematicii*. Ed. Albatros, Bucureşti, 1987.
37. *Moduri de gândire*. Colecţia “Ştiinţa pentru toţi", Ed. Ştiinţifică şi Enciclopedică, Bucureşti, 1987.
38. *Provocarea ştiinţei*. Seria “Idei contemporane", Ed. Politică, Bucureşti, 1988.
39. *Invenţie şi descoperire*. Ed. Cartea Românească, 1989.
40. *Analiza matematică. Materiale pentru perfecţionarea profesorilor de liceu III*. Universitatea din Bucureşti, Facultatea de Matematică, Bucureşti, 1989 (co-author).
41. *Dicţionar de Analiză Matematică*. Ed. Ştiinţifică şi Enciclopedică, Bucureşti, 1989 (co-author).
42. *Controverse în ştiinţă şi inginerie*. Ed. Tehnică, Bucureşti, 1991.
43. *Language, Logic, Cognition and Communication; A Semiotic,Computational and Historical Approach*. Report 9/96. Grup de Recerca en Linguistica Matematica i Enginyeria del Llenguatge. Reports Universitat Rovira i Virgili, Tarragona, Spain, 1996.
44. *Matematica, manual pentru clasa a IX-a, licee teoretice specializarea filologie* (in colab. with Mihaela Singer). Ed. Sigma, Bucureşti, 1999; 2nd edition, 2000.
45. *Matematica, manual pentru clasa a XII-a* (in colab. with P. Alexandrescu, M. Radulescu, S. Radulescu). Ed. Paralela 45, Bucureşti – Piteşti, 2002.
46. *Jocul ca libertate*. Ed. Scripta (Colecţia ludică), Bucureşti, 2003.
47. *Mathematics in Romania*. CUB Press 22, Baia Mare, 2004.
48. *Intâlnirea extremelor. Scriitori în orizontul ştiinţei*. Ed. Paralela 45, Bucureşti – Piteşti, 2005.
49. *Paradigme universale*. Ed. Paralela 45, Piteşti – Bucureşti, 2005.
50. *Paradigme universale II. Pornind de la un zâmbet*. Ed. Paralela 45, Piteşti – Bucureşti, 2006.
51. *Paradigme universale III. Jocul*. Ed. Paralela 45, Piteşti – Bucureşti, 2007.
52. *Words and Languages Everywhere*. Polimetrica Academic Publisher, Milano, 2007.
53. *Solomon Marcus (protagonist): Educaţia în spectacol*. Spandugino Publ. House., Bucureşti, 2010.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 2, pp. 148-159

Middleware for Smart Environments Management

I. Anghel, T. Cioara, I. Salomie, M. Dinsoreanu, A. Rarau

Ionut Anghel, Tudor Cioara, Ioan Salomie, Mihaela Dinsoreanu, Anca Rarau
Technical University of Cluj-Napoca
Computer Science Department
15 Daicoviciu street, Cluj-Napoca, Romania
E-mail: {ionut.anghel,tudor.cioara,ioan.salomie,mihaela.dinsoreanu,anca.rarau}@cs.utcluj.ro

Abstract: This paper introduces a self-configuring middleware that manages the processes of context information acquisition and representation from smart closed environments, targeting the development of context aware applications. The environment context information is modeled using three sets: context resources, context actors and context policies. The context model artifacts are generated and administrated at run time by a management infrastructure based on intelligent software agents. The self-configuring property is enforced by monitoring the closed environment in order to detect variations or conditions for which the context model artifacts must be updated. The middleware was tested and validated within the premises of our Distributed Systems Research Laboratory smart environment.

Keywords: Self-Configuring, Closed Spaces, Context Awareness, Autonomic Computing, Pervasive Systems

1 Introduction and Motivation

The context aware applications continuously monitor, capture and interpret environment related information in order to adapt their behavior to changes. A context aware system can evolve in two distinct types of environments: open environments and closed environments. An open environment is a large, heterogeneous space with no fixed spatial limitations (e.g. a city or an airport). In an open space the most relevant context information is the location which can be determined through GPS devices. On the other hand, a closed environment is a smart space delimited by predefined fixed boundaries featuring a large number of sensors and intelligent devices (e.g. a smart hospital) where the actor's location is identified by indoor specific techniques such as RFID (Radio-Frequency Identification) [1] or location sensors. These spaces offer a large variety of context information which is usually captured through sensor networks.

Let's consider a scenario in which a context aware application is used to guide the tourists into a museum. The museum is an intelligent closed space in which visitors are identified by RFID readers, while their location and orientation is determined using a sensor network. The tourists, as scenario actors, can use the context aware application if they have wireless capable PDAs on which an application can be downloaded and executed. In the museum, the visitors must follow a set of rules such as the minimum distance to the artifacts, loud limits, etc. The interaction between the tourists and the museum resources is made through requests from the tourists to the application residing on their PDAs. These requests may vary in complexity from simple requests such as information about a specific artifact, to complex requests such as requesting a tour to visit the middle age artifacts. Using this scenario, the following environment resources which provide relevant information and determine the context aware application execution are identified: the resources attached to the tourist such as PDA or RFID tags; the intelligent museum resources such as location sensors, orientation sensors or actuators; the constraints used to drive the tourist-museum interaction such as the minimum distance to the artifacts; the tourist preferences and his mobility.

The resources variety may influence the execution of a context aware application making the context information acquisition and representation processes extremely difficult to manage. During the context information acquisition process, the context information sources (e.g. sensors) can fail or new context information sources may be identified. The context acquisition and representation processes need to be reliable and fault tolerant. A context aware application cannot wait indefinitely for an answer from a temporary unavailable context resource. On the other hand, the payoff for not taking into consideration the new available context resources can be very high.

From our perspective it is necessary to introduce some degree of autonomy to the context acquisition and representation processes in order to achieve an efficient closed space context information management. We address the problems of monitoring and capturing context information by proposing and developing a pervasive self-configuring middleware that manages intelligent closed space environments. The fundamental element of this middleware is the context model which represents the environment context information using three sets: context resources, context actors and context policies. The context model management infrastructure is implemented by intelligent software agents [2] that generate and administrate the context model artifacts at run time. The middleware self-configuring feature is implemented by monitoring and evaluating the environment changes in order to keep the context artifacts updated. The proposed middleware is tested and validated in our Distributed Systems Research Laboratory [3] smart environment infrastructure.

The rest of the paper is organized as follows: in Section 2, a domain research overview is presented; Section 3 presents the middleware conceptual architecture; Section 4 presents the middleware self-configuring feature and a self-configuring algorithm; Section 5 shows how the middleware is used to manage the context representation of a smart laboratory while Section 6 concludes the paper and shows the future work.

2 Related Work

During the middleware development process we have identified the context representation and the autonomic context management as major problems. The rest of this section presents the state of the art related to these research directions by highlighting our approach advantages.

For *context representation*, generic models that aim at accurately describing the system execution context in a programmatic way are proposed [4]. Key-Value models represent context information using a set of attributes and their associated values [5]. Markup models enable structuring context information into a hierarchy. Tags describe context attributes and associated values [6]. Object models structure context into object classes and their implicit relationships [7]. In [8], the concept of multi-faceted entity is defined and used to model the set of context properties. A facet represents the effective values of the context properties to which the context aware application has access. The main drawback of the above presented approaches is the lack of semantic information encapsulated in the context model representation which makes the process of inferring new context related knowledge extremely difficult. In the current work we try to overcome these deficiencies by using our RAP context model [9] to represent the context information. The RAP context model ontology based representation is used by the context aware applications to infer new context related information using reasoning and learning algorithms. The use of ontology representations to model the context related information is also proposed in [10], [11]. In this approaches the context properties are represented as ontological concepts during design time and instantiated with run-time sensor captured values. The main disadvantage of these approaches is the high degree of inflexibility determined by the human intervention in the context representation phase. This problem is solved in our approach by defining an agent based solution that uses the RAP context model set based context representation to evaluate the real world context and to automatically construct the context representation.

In the *context management direction*, the research is concentrated on developing techniques for (i) keeping the context representation consistent with the environment and for (ii) automatic discovery and setup of new context resources. In [12], models for capturing and updating the context information based on the information type are proposed. In this approach the context information is classified in three types: user information, physical information and computational information. Another approach is to define reusable components for updating the context specific data [4]. These components provide stable communication channels for capturing and controlling context specific data. In [13] the authors propose the development of context guided behavioral models, which allow context aware applications to detect only those context data variations that lead to the modification of their behavior. The main disadvantage of these approaches is the lack of efficiency in the context model management process which is rather static and difficult to adapt to context changes. The current approach introduces a degree of autonomy for the context acquisition and representation processes in order to provide an efficient context information management. There is a reduced number of researches regarding the context aware systems self-* management in the literature and they are focused on the self-adaptation problem. Models and algorithms that allow computational systems to execute specific actions according to the context or situation at hand are proposed [14]. Their objective is to associate a certain degree of intelligence to the computational systems for context adaptation. In [15], the authors propose a context adaptive platform based on the closed loop control principle. The novelty of this proposal consists in defining and using the concept of application-context description to represent system knowledge about the context. This description is frequently updated and used for system control allowing it to reconfigure and take adapting decisions. [16], [17] and [18] propose context adaptation models for web services and web service orchestrations based on defining the service behavior in a certain situation using a set of context adapting rules, each rule consisting of a context condition and an associated action. A self-adapting context model that uses a system situation space to represent the system execution context is proposed in [19]. Using learning algorithms, the system may infer the action to be executed when a new situation appears by placing it in a situation space group. The use of the self-configuring autonomic computing paradigm for the development and integration of self-managing components into context sensitive systems is a new research direction. The research efforts concentrate on developing models for managing the automatic discovery, installation and configuration of complex context aware systems and their components [20], [21]. In this paper we make a step forward towards developing context aware self-* management systems by defining a self-configuring algorithm that is used to develop a context aware management middleware for smart environments.

3 The Middleware Conceptual Architecture

The middleware architecture (see Figure 1) defines three main layers: the *acquisition layer* that captures the context information from closed environments, the *context model layer* which represents the context information in a machine interpretable way and the *context model management layer*. In the following we detail each of the three middleware architectural layers.

3.1 The Context Acquisition Layer

The middleware monitors and captures information form the closed environment in order to accurately represent the context. Our approach to the context information acquisition layer (see Figure 2) provides (i) a sensor information acquisition mechanism and (ii) easy access and good visibility of the sensor information to middleware upper layers.

From the middleware perspective we have defined both push and pull types of sensor information acquisition mechanisms. The push mechanism uses event based listeners to determine when information is available in order to make it visible to the middleware upper layers. The pull mechanism is query

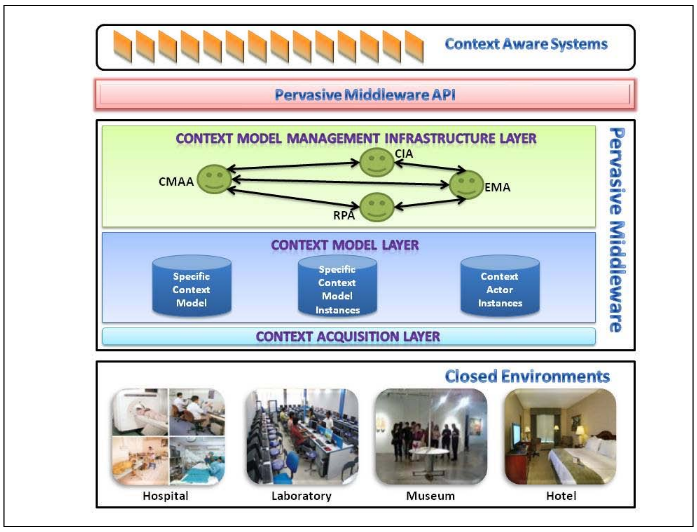


Figure 1: The Pervasive Middleware Conceptual Architecture

based, allowing the sensor information to be provided on demand. The sensor information is made visible to the upper middleware layers by the exposing associated web services.

3.2 The Context Model Layer

To represent a closed environment context information in a programmatic manner we have used the RAP context model [9]. In RAP, the context is defined as a triple: $C = <R, A, P>$, where R is the set of context resources that generates and/or processes context information, A is the set of actors which interact with context resources in order to satisfy their needs and P is the set of real world context related policies. The set of context resources R is split in two disjunctive subsets: (i) the set of context resources attached to the real world context environment R_S and (ii) the set of context resources attached to the actors R_A.

In order to provide an accurate representation of a closed environment context, the following representation artifacts are defined (see Figure 3): *specific context model*, *specific context model instance* and *context - actor instance*. The specific context model $C_S = <R_S, A_S, P_S>$ is obtained by mapping the context model onto different closed environments and populating the sets with environment specific elements. A specific context model instance $C_{SI} = <R_{SI}, A_{SI}, P_{SI}>$ contains the set of context resources with which the middleware interacts, together with their values in a specific moment of time t. The context - actor instance $CI_a{}^t = < R_a{}^t, a, P^t >$ contains the set of context resources with which the actor can interact, together with their values in a specific moment of time t.

3.3 The Context Model Management Layer

The context model management infrastructure layer is based on four types of intelligent, cooperative intelligent software agents: *Context Model Administering Agents*, *Context Interpreting Agents*, *Request Processing Agents* and *Execution and Monitoring Agents*. The Context Model Administering Agent (CMAA) is the specific context model manager. Its main goal is the synchronization of the context model artifacts with the system execution environment. It is also responsible for negotiation processes that take place when an actor or resource is joining the context. The Context Interpreting Agent (CIA) semantically evaluates the information of a context instance and tries to find the context instance meaning. The Request Processing Agent (RPA) processes the actor requests. RPA identifies and generates the action plans that must be executed for serving an incoming request. Also, RPA uses the context

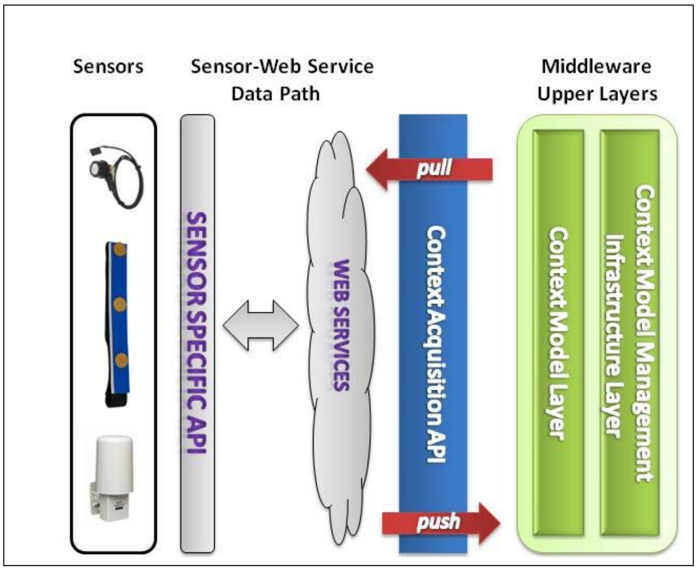


Figure 2: The context acquisition layer

model artifacts to identify the proper plan to be executed by the Execution and Monitoring Agent or for generating a new plan. The Execution and Monitoring Agent (EMA) processes the plans received from RPA agent and executes every plan action using the available services. After mapping action plans onto services, a plan orchestration which can be executed using transactional principles is obtained.

4 The Self-Configuring Feature

In order to provide an efficient, reliable and fault tolerant closed space context information management, the middleware is enhanced with the autonomic computing self-configuring property. This property is enforced by: (i) monitoring the closed environment in order to detect context variations or conditions for which the context artifacts must be updated (section 4.1.) and (ii) executing a self-configuring algorithm (section 4.2.) that updates/populates the context artifacts.

4.1 The Closed Environment Context Variation

We have identified three causes that generate context variation: (1) adding or removing context elements (resources, actors, policies) to/from the closed environment, (2) the actors' mobility within the environment and (3) the changes of the resources' property values (mainly due to changing the sensors' captured values). In the following, we discuss each of these context variation causes to evaluate the context variation degree which determines the condition of executing the self-configuring process.

Adding or removing context elements. During the context information acquisition process, the sources of context information can fail or randomly leave/join the context. These changes generate a context variation that is detected by the context acquisition layer and sent to CMAA which creates context artifacts adapted to the environment. Next, we evaluate the context variation degree generated by context resources ΔR, context actors ΔA and context policies ΔP in relationship with a set of associated thresholds T_R, T_A and T_P respectively. The context resources set variation is generated by adding or removing a context resource *r* (sensor or actuator) to/from the closed environment:

$$\Delta R = \{ R^{t+1} \setminus R^{t} \} \cup \{R^{t} \setminus R^{t+1}\} \quad (1)$$

In relation (1) $R^{t+1} \setminus R^{t}$ contains the set of context resources that become available and $R^{t} \setminus R^{t+1}$ contains

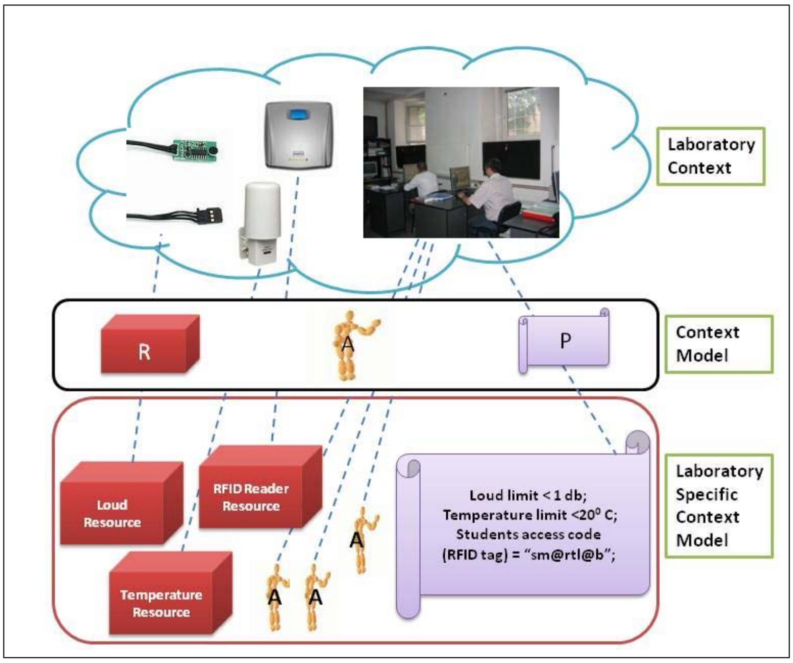


Figure 3: The RAP context model

the set of context resources that become unavailable. If Card(ΔR) $\geq$ T$_R$ a new specific context model is generated by adding or removing the context resources contained in ΔR. Using the same assumptions and conclusions as for context resources, we compute the policy set variation:

$$\Delta \mathrm{P} = \{ \mathrm{P}^{t+1} \setminus \mathrm{P}^{t} \} \ \mathrm{U} \ \{\mathrm{P}^{t} \setminus \mathrm{P}^{t+1}\} \qquad (2)$$

For evaluating the actors related context, the fact that each context actor has a set of attached resources was taken into consideration:

$$\Delta \mathrm{A} = \{ \mathrm{A}^{t+1} \setminus \mathrm{A}^{t} \} \ \mathrm{U} \ \{\mathrm{A}^{t} \setminus \mathrm{A}^{t+1}\} \ \mathrm{U} \ \{ \mathrm{R}_A{}^{t+1} \setminus \mathrm{R}_A{}^{t} \} \ \mathrm{U} \ \{\mathrm{R}_A{}^{t} \setminus \mathrm{R}_A{}^{t+1}\} \qquad (3)$$

The overall real world context variation ΔENV is given by the union of all context elements' variation:

$$\Delta \mathrm{ENV} = \Delta \mathrm{R} \ \mathrm{U} \ \Delta \mathrm{A} \ \mathrm{U} \ \Delta \mathrm{P} \qquad (4)$$

CMMA should start the execution of the self-configuring process and should generate a new specific context model when Card(ΔENV) $\geq$ T$_{Self-Configuring}$, where the self-configuring threshold is defined as:

$$\mathrm{T}_{Self-Configuring} = \min(\mathrm{T}_R, \mathrm{T}_A, \mathrm{T}_P) \qquad (5)$$

The actor's mobility. Due to their mobility, the actors are changing their environment location and implicitly the set of resources with which they interact. CMAA identifies this variation and generates (i) a new context-actor instance and (ii) a new specific context model instance. In order to evaluate the context variation generated by actors' mobility we use the isotropic context space concept, defined in [9]. A context space is isotropic if and only if the set of real world context resources is invariant to the actors' movement. Usually, a context space is non-isotropic, but it can be split into a set of disjunctive isotropic context sub-space volumes in which the isotropy degree variation is the empty set. Such volume is called context granule. For a given moment of time, an actor can be located only into a single context granule. As a result, the space isotropy variation ΔIZ is non-zero only when an actor a moves between two context granules. The isotropy variation for a context actor is computed as:

$$\Delta IZ_a = \{ R_{GC}{}^{t+1} \setminus R_{GC}{}^{t} \} \cup \{R_{GC}{}^{t} \setminus R_{GC}{}^{t+1}\} \qquad (6)$$

CMMA continuously monitors the actors' movement in the environment and periodically evaluates the space isotropy variation. If for an actor, the space isotropy variation is a non empty set, $\Delta IZ_a \neq \emptyset$, then the self-configuring process executed by CMMA generates a new context-actor instance:

$$CI_a{}^t = \, < R_a{}^t, a, P^t >, R_a{}^{t+1} = R_{GC}{}^{t+1} \qquad (7)$$

Changes of resources property values. A context resource is a physical or virtual entity which generates and/or processes context information. The resource properties, K(r), specify the set of relevant context information that a resource can provide. For example, the set of context properties for a HotandHumidity sensor is: K(HotandHumidity) = {Temperature, Humidity}. In order to evaluate the context variation generated by the changes of the resource property values, we define a function K_{val} that associates the resource property to its value:

$$K_{val}(R) = \{(k_1,val_1), \ldots, (k_n,val_n)\} \text{ with } k_1, \ldots, k_n \in K \qquad (8)$$

If the values captured by the HotandHumidity sensor in a moment of time are 5 degree Celsius for temperature and 60% for humidity, then: K_{val}(HotandHumiditySensor) = {(Temperature, 5), (Humidity, 60%)}. The CMAA agent calculates the context variation generated by changes of resource properties' values ΔRPV as presented in (9). If val^{t+1}- val^t=0 then the property value hasn't changed from t to t+1 and that property is ignored when the variation is calculated. As a result, we conclude that a new specific context model instance should be created when Card(ΔRPV) ≥ 0.

$$\Delta RPV = K_{val}(R^{t+1}) - K_{val}(R^t) = \{(k_1,val_1{}^{t+1} - val_1{}^t), \ldots, (k_n,val_n{}^{t+1} - val_n{}^t)\} \qquad (9)$$

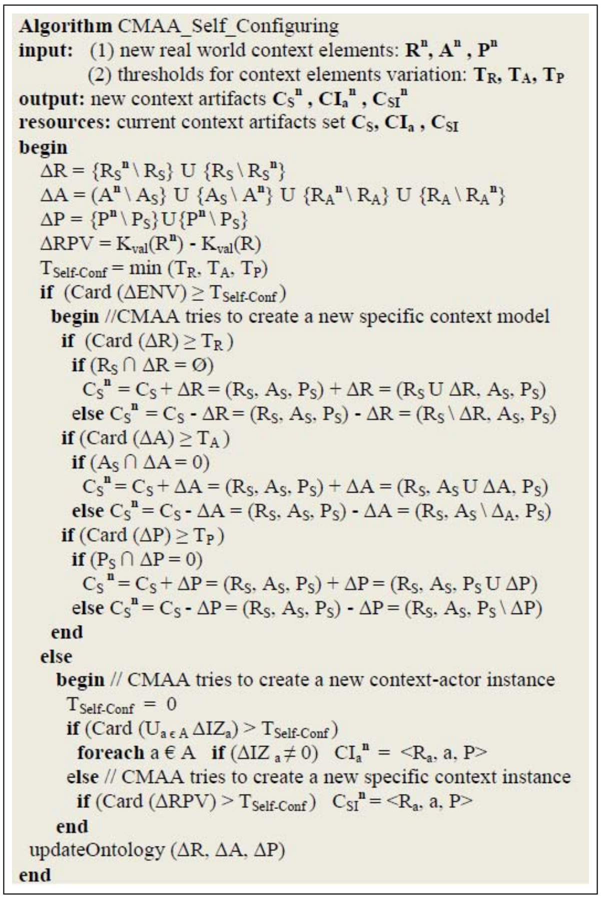
4.2 The Self-Configuring Algorithm

The self-configuring algorithm is executed by CMAA in order to keep the context model artifacts synchronized with the real context (see Figure 4). CMAA features a ticker based behavior by periodically evaluating the context changes. When a significant context variation is determined, the context model artifacts are updated.

5 Case Study

For the case study we have used the intelligent closed environment represented by our Distributed System Research Laboratory. In the laboratory students are marked using RFID tags and identified using a RFID reader. The students interact with the smart laboratory by means of wireless capable PDAs on which different laboratory provided services are executed (submit homework service, print services, information retrieval services, etc.). A sensor network captures information regarding students' location or orientation and also information regarding the ambient like the temperature or humidity. The DSRL infrastructure contains a set of sensors through which the real context information is collected: two HotandHumidity sensors that capture the air humidity and the temperature, four Orientation sensors placed in the four corners of the laboratory that measure the orientation on a single axis, one Loud sensor that detects sound loudness level and one Far Reach sensor that measures distances (see Figure 5).

The sensors are connected using a Wi-microSystem wireless network produced by Infusion Systems Ltd [22]. The middleware is deployed on an IBM Blade-based technology Server Center which maintenance software offers autonomic features like self-configuring of its hardware resources. The context related data captured by sensors is collected through the Wi-microSystem that has an I-CubeX WimicroDig analog to digital encoder as its main part. It is a configurable hardware device that encodes up to



```
Algorithm CMAA_Self_Configuring
input:  (1) new real world context elements: $R^n$, $A^n$, $P^n$
        (2) thresholds for context elements variation: $T_R$, $T_A$, $T_P$
output: new context artifacts $C_S^n$, $CI_a^n$, $C_{SI}^n$
resources: current context artifacts set $C_S$, $CI_a$, $C_{SI}$
begin
   $\Delta R = \{R_S^n \setminus R_S\} \cup \{R_S \setminus R_S^n\}$
   $\Delta A = (A^n \setminus A_S\} \cup \{A_S \setminus A^n\} \cup \{R_A^n \setminus R_A\} \cup \{R_A \setminus R_A^n\}$
   $\Delta P = \{P^n \setminus P_S\} \cup \{P^n \setminus P_S\}$
   $\Delta RPV = K_{val}(R^n) - K_{val}(R)$
   $T_{Self\text{-}Conf} = \min(T_R, T_A, T_P)$
   if $(Card(\Delta ENV) \geq T_{Self\text{-}Conf})$
    begin //CMAA tries to create a new specific context model
     if $(Card(\Delta R) \geq T_R)$
      if $(R_S \cap \Delta R = \emptyset)$
        $C_S^n = C_S + \Delta R = (R_S, A_S, P_S) + \Delta R = (R_S \cup \Delta R, A_S, P_S)$
      else $C_S^n = C_S - \Delta R = (R_S, A_S, P_S) - \Delta R = (R_S \setminus \Delta R, A_S, P_S)$
     if $(Card(\Delta A) \geq T_A)$
      if $(A_S \cap \Delta A = 0)$
        $C_S^n = C_S + \Delta A = (R_S, A_S, P_S) + \Delta A = (R_S, A_S \cup \Delta A, P_S)$
      else $C_S^n = C_S - \Delta A = (R_S, A_S, P_S) - \Delta A = (R_S, A_S \setminus \Delta_A, P_S)$
     if $(Card(\Delta P) \geq T_P)$
      if $(P_S \cap \Delta P = 0)$
        $C_S^n = C_S + \Delta P = (R_S, A_S, P_S) + \Delta P = (R_S, A_S, P_S \cup \Delta P)$
      else $C_S^n = C_S - \Delta P = (R_S, A_S, P_S) - \Delta P = (R_S, A_S, P_S \setminus \Delta P)$
    end
   else
     begin // CMAA tries to create a new context-actor instance
      $T_{Self\text{-}Conf} = 0$
      if $(Card(\cup_{a \in A} \Delta IZ_a) > T_{Self\text{-}Conf})$
       foreach $a \in A$  if $(\Delta IZ_a \neq 0)$  $CI_a^n = <R_a, a, P>$
      else // CMAA tries to create a new specific context instance
       if $(Card(\Delta RPV) > T_{Self\text{-}Conf})$  $C_{SI}^n = <R_a, a, P>$
     end
  updateOntology $(\Delta R, \Delta A, \Delta P)$
end
```

Figure 4: The self-configuring algorithm

8 analog sensor signals to MIDI messages which are real-time wirelessly transmitted, through Bluetooth waves, to the Server Center for analysis and/or control purposes. The Bluetooth receiver located on the Blade computer is mapped as a Virtual Serial Port.

CMAA periodically evaluates the context information changes at a predefined time interval (we use 1 second time intervals for this purpose). If significant variations are detected, the context model artifacts are created or updated using the self-configuring algorithm presented in section 4.2. When the middleware is deployed and starts its execution (t=0), there are no context model artifacts constructed meaning that the R, A and P sets of the context model are empty. After one second (t=1), when two students John and Mary enter the lab, CMAA receives the updated context information from the Context Acquisition Layer and calculates the context elements variation ΔR, ΔA and ΔP as presented in Figure 6. By default the self-configuring thresholds are set to the value 1 ($T_{Self-Configuring}$=T_R=T_A=T_P=1). As a result of evaluating the context variation at t=1, CMAA executes the self-configuring algorithm which adds new concepts/populates the context model artifacts. The new added concepts originate from the context elements set variations ΔR, ΔA and ΔP calculated in Figure 6.

In order to assess the performance of the proposed self-configuring algorithm a simulation editor was developed. The evaluation test cases can be described by adding simulation times together with the corresponding sensor values. We evaluated the memory and processor overloading when executing the self-configuring algorithm in order to update the specific context model instance due to sensor values changes. Using the simulator, we tested our middleware with 100 sensors that change their values at 100 ms and 2000 ms. Even if the sensor values change rate is much higher at 2000 ms, the memory and processor overloading did not show major differences (see Figure 7).

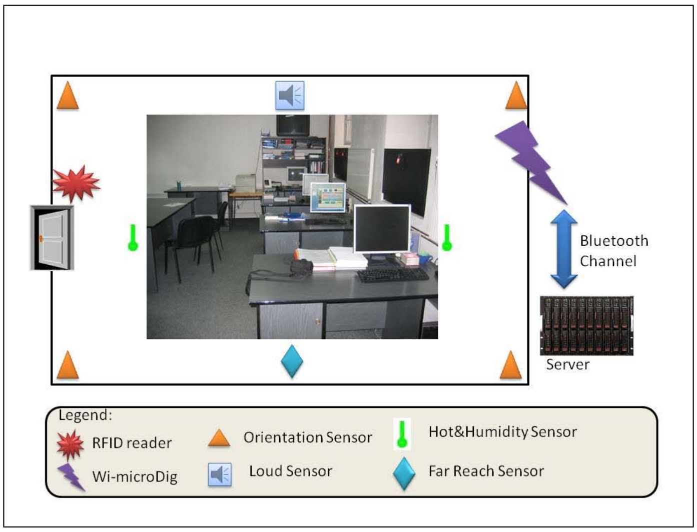
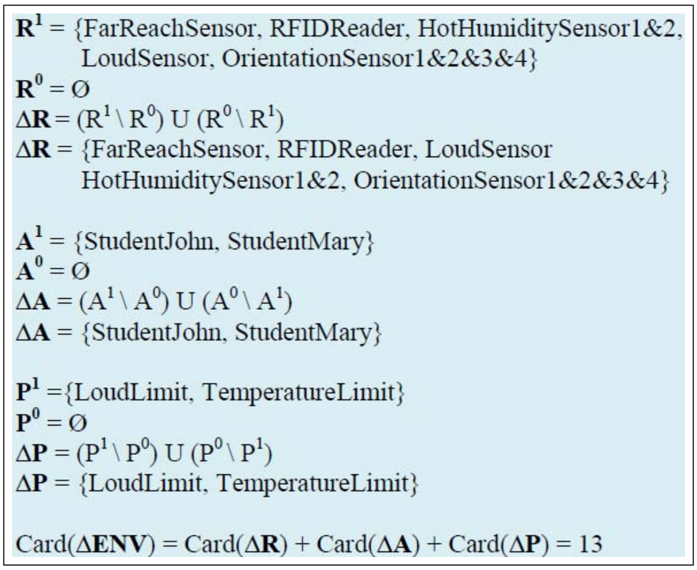


Figure 5: The DSRL infrastructure



$\mathbf{R}^1$ = {FarReachSensor, RFIDReader, HotHumiditySensor1&2, LoudSensor, OrientationSensor1&2&3&4}

$\mathbf{R}^0 = \emptyset$

$\Delta\mathbf{R} = (R^1 \setminus R^0)\ \mathrm{U}\ (R^0 \setminus R^1)$

$\Delta\mathbf{R}$ = {FarReachSensor, RFIDReader, LoudSensor HotHumiditySensor1&2, OrientationSensor1&2&3&4}

$\mathbf{A}^1$ = {StudentJohn, StudentMary}

$\mathbf{A}^0 = \emptyset$

$\Delta\mathbf{A} = (A^1 \setminus A^0)\ \mathrm{U}\ (A^0 \setminus A^1)$

$\Delta\mathbf{A}$ = {StudentJohn, StudentMary}

$\mathbf{P}^1$ ={LoudLimit, TemperatureLimit}

$\mathbf{P}^0 = \emptyset$

$\Delta\mathbf{P} = (P^1 \setminus P^0)\ \mathrm{U}\ (P^0 \setminus P^1)$

$\Delta\mathbf{P}$ = {LoudLimit, TemperatureLimit}

$\mathrm{Card}(\Delta\mathbf{ENV}) = \mathrm{Card}(\Delta\mathbf{R}) + \mathrm{Card}(\Delta\mathbf{A}) + \mathrm{Card}(\Delta\mathbf{P}) = 13$

Figure 6: DSRL context variation at t=1

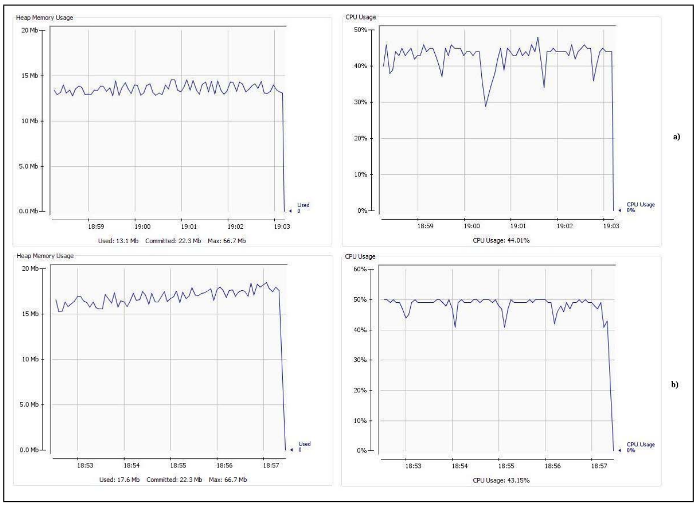


Figure 7: The self-configuring algorithm CPU and memory overloading with 100 sensors at a) t1=100 ms and b) t2=2000 ms

For testing our self-configuring algorithm scalability we have implemented an application that can simulate the behavior of a large number of sensors that randomly generate context information at fixed periods of time. The results show that the self-configuring algorithm implemented by CMAA can generate, synchronize and update the context model artifacts in a reasonable time for up to 20 sensors that change their values simultaneously. It is possible that sensor values change much faster than CMAA can synchronize contexts when the processing time is higher than the CMAA ticker interval.

6 Conclusions

This paper addresses the problem of managing the closed environment context information acquisition and representation processes in a reliable and fault tolerant manner. In order to achieve our goal we have defined a self-configuring middleware that uses a context management infrastructure based on agents, to gather context information from sensors and generate a context representation at run-time. The self-configuring property is enforced at the middleware level by monitoring the execution context in order to detect context variations for which the context artifacts must be updated. As a result, the middleware supports the dynamic configuration of the context artifacts. For future development, we intend to enhance the middleware with self-healing and self-optimizing autonomic features.

Bibliography

[1] G. Roussos and V. Kostakos, rfid in pervasive computing: State-of-the-art and outlook, *Pervasive and Mobile Computing*, Volume 5, Issue 1 pp. 110-131, ISSN:1574-1192, 2009.

[2] I. Dzitac and B. E. Barbat, Artificial Intelligence + Distributed Systems = Agents, *International Journal of Computers, Communications and Control (IJCCC)*, Vol 4(1), pp. 17-27, ISSN: 1841-9836, 2009.

[3] Distributed Systems Research Laboratory, Technical University of Cluj-Napoca, *http://dsrl.coned.utcluj.ro*.

[4] D. Fournier, S. Ben Mokhtar and N. Georgantas, Towards Ad hoc Contextual Services for Pervasive Computing, *Proceedings of the 1st workshop on Middleware for Service Oriented Computing*, pp. 36 - 41, ISBN:1-59593-425-1, 2006.

[5] K. M. Anderson, F. A. Hansen and N. O. Bouvin, Templates and queries in contextual hypermedia, *Proceedings of the seventeenth conference on Hypertext and hypermedia*, pp. 99 - 110, 2006.

[6] D. Raz, A. Tapani Juhola, J. Serrat-Fernandez and A. Galis, *Fast and Efficient Context-Aware Services*, Wiley Series on Communications Networking and Distributed Systems, ISBN-13: 978-0470016688, pp. 5-25, 2006.

[7] T. Hofer, W. Schwinger, M. Pichler, G. Leonhartsberger, J. Altmann and W. Retschitzegger, Context-awareness on mobile devices - the hydrogen approach, *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, pp. 292, USA, 2003.

[8] A. Rarau, K. Pusztai and I.Salomie, MultiFacet Item based Context-Aware Applications, *Inernational Journal of Computer and Information Sciences 3(2)*, pp. 10-18, 2006.

[9] I. Salomie, T. Cioara, I. Anghel and M. Dinsoreanu, RAP - A Basic Context Awareness Model, *Proc. Of 4th IEEE Int. Conf. on Intelligent Computer Communication and Processing*, ISBN: 978-1-4244-2673-7, pp. 315- 318, 2008.

[10] G. Specht and T. Weithoner, Context-Aware Processing of Ontologies in Mobile Environments, *7th Int. Conference on Mobile Data Management*, ISBN: 978-1-4244-2673-7, pp. 86, May 2006.

[11] K. C. Lee, J. H. Kim and J. H. Lee, Implementation of Ontology Based Context-Awareness Framework for Ubiquitous Environment, *Int. Conference on Multimedia and Ubiquitous Engineering*, pp. 278 - 282, April 2007.

[12] P. Bellavista, A. Corradi and R. Montanari, Mobile Computing Middleware for Location and Context-Aware Internet Data Serices, *ACM Transactions on Internet Technology (TOIT)*, Volume 6 , Issue 4, pp. 356 - 380, ISSN:1533-5399, 2006.

[13] G. Spanoudakis and K. Mahbub, A Platform for Context Aware Runtime Web Service Discovery, *IEEE Int. Conference on Web Services*, pp. 233-240, ISBN: 0-7695-2924-0, 2007.

[14] C. Klein, R. Schmid, C. Leuxner, W. Sitou and B. Spanfelner, A Survey of Context Adaptation in Autonomic Computing, *Proc. of the Fourth International Conference on Autonomic and Autonomous Systems*, pp. 106 - 111, 2008.

[15] M. Cremene, M. Riveill and C. Martel, A Service-Context Model allowing Dynamical Adaptation, *International Journal of Computers, Communications and Control (IJCCC)*, Vol 1, pp. 163-170, ISSN: 1841-9836, 2006.

[16] F. Seyler, C. Taconet and G. Bernard, Context Aware Orchestration Meta-Model, *Proc. Of the Third Int. Conference on Autonomic and Autonomous Systems*, pp. 17, 2007.

[17] Y. C. Zhou, X. P. Liu, E. Kahan, X. N. Wang, L. Xue and K. X. Zhou, Context Aware Service Policy Orchestration, *Proc of the IEEE International Conference on Web Services*, pp. 936-943, 2007.

[18] I. Y.L. Chen, S. J.H. Yang and J. Zhang, Ubiquitous Provision of Context Aware Web Services, *IEEE International Conference on Services Computing (SCC'06)*, pp.60-68, ISBN: 0-7695-2670-5, 2006.

[19] N. O'Connor, R. Cunningham and V. Cahill, Self-Adapting Context Definition, *First International Conference on Self-Adaptive and Self-Organizing Systems*, pp. 336-339, 2007.

[20] R. Calinescu, Model-Driven Autonomic Architecture, *Proc. of the Fourth International Conference on Autonomic Computing*, pp. 9, 2007.

[21] E. Patouni and N. Alonistioti, A Framework for the Deployment of Self-Managing and Self-Configuring Components in Autonomic Environments, *Proc. of the Int. Symposium on a World of Wireless, Mobile and Multimedia Networks*, pp. 484-489, 2006.

[22] Infusion Systems Ltd, *www.infusionsystems.com*.

Ionut Anghel (February 28, 1983) is an Assistant Professor and PhD Student at the Computer Science Department of the Technical University of Cluj-Napoca. He is currently developing the PhD thesis "Autonomic Computing Middleware". His main research areas are Autonomic systems, Mobile agents and Ontology learning. Also he was involved in four national funded research projects and published several papers in international databases-indexed proceedings and journals in the last 2 years. Ionut also co-authored a book in distributed systems in 2008.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 2, pp. 160-170

EWMA Algorithm in Network Practice

P. Čisar, S. Bošnjak, S. Maravić Čisar

Petar Čisar
Telekom Srbija
Prvomajska 2-4, Subotica, Serbia
E-mail: petarc@telekom.rs

Saša Bošnjak
Faculty of Economics
Segedinski put 9-11, Subotica, Serbia
E-mail: bsale@eccf.su.ac.yu

Sanja Maravić Čisar
Subotica Tech
Marka Oreškovića 16, Subotica, Serbia
E-mail: sanjam@vts.su.ac.rs

Abstract: Abstract: Intrusion detection is used to monitor and capture intrusions into computer and network systems which attempt to compromise their security. Many intrusions manifest in changes in the intensity of events occuring in computer networks. Because of the ability of exponentially weighted moving average (EWMA) control charts to monitor the rate of occurrences of events based on their intensity, this technique is appropriate for implementation in control limits based algorithms. The paper also gives a review of a possible optimization method. The validation check of results will be performed on authentic network samples.

Keywords: intrusion detection, EWMA, control limits, optimization, autocorrelation

1 Introduction

The exponentially weighted moving average is a statistic for monitoring the process that averages the data in a way that gives less and less weight to data as they are further removed in time. For the EWMA control technique, the decision regarding the state of control of the process depends on the EWMA statistic, which is an exponentially weighted average of all prior data, including the most recent measurements.

By the choice of weighting factor λ, the EWMA control procedure can be made sensitive to a small or gradual drift in the process.

The statistic that is calculated is the following:

$$EWMA_t = \lambda Y_t + (1-\lambda)EWMA_{t-1} \qquad t = 1,2,\ldots,n \tag{1}$$

where

- $EWMA_0$ is the mean of historical data (target)
- Y_t is the observation at time t
- n is the number of observations to be monitored including EWMA_0
- $0 < \lambda \leq 1$ is a constant that determines the depth of memory of the EWMA.

This equation has been established by Roberts as described in [4].

The parameter λ determines the rate at which "older" data enter into the calculation of the EWMA statistic. A value of λ = 1 implies that only the most recent measurement influences the EWMA. Thus, a large value of λ = 1 gives more weight to recent data and less weight to older data - a small value of λ gives more weight to older data. The value of λ is usually set between 0.2 and 0.3 [2] although this choice is somewhat arbitrary. Lucas and Saccucci [3] have shown that although the smoothing factor λ used in an EWMA chart is usually recommended to be in the interval between 0.05 to 0.25, in practice the optimally designed smoothing factor depends not only on the given size of the mean shift δ, but also on a given in-control Average Run Length (ARL).

The estimated variance of the EWMA statistic is approximately:

$$\sigma^2_{EWMA} = \frac{\lambda}{2-\lambda} \cdot \sigma^2 \tag{2}$$

where σ is the standard deviation calculated from the historical data.

The center line for the control chart is the target value or $EWMA_0$. The upper and lower control limits are:

$$\begin{aligned} UCL &= EWMA_0 + k\sigma_{EWMA} \\ LCL &= EWMA_0 - k\sigma_{EWMA} \end{aligned} \tag{3}$$

where the factor k is either set equal 3 (the 3-sigma control limits) or chosen using the Lucas and Saccucci tables (ARL = 370).

In addition to the aforementioned authors, the publications [6]-[13] have also dealt with the topic of EWMA statistics and statistical anomaly detection in computer networks.

Control charts are specialized time series plots, which assist in determining whether a process is in statistical control. Some of the most widely used forms of control charts are X-R charts and Individuals charts. These are frequently referred to as "Shewhart" charts after the control charting pioneer Walter Shewhart who introduced such techniques. These charts are sensitive to detecting relatively large shifts in the process (i.e. of the order of 1.5σ or above). In computer network practice, shifts can be caused by intrusion or attack, for example. Two types of charts are primarily used to detect smaller shifts (less than 1.5σ), namely cumulative sum (or CUSUM) charts and EWMA charts. A CUSUM chart plots the cumulative sums of the deviations of each sample value from a target value. An alternative technique to detect small shifts is to use the EWMA methodology. This type of chart has some very attractive properties, in particular:

1. Unlike X-R and Individuals charts, all of the data collected over time may be used to determine the control status of a process.

2. Like the CUSUM, the EWMA utilizes all previous observations, but the weight attached to data exponentially decreases as the observations become older and older.

3. The EWMA is often superior to the CUSUM charting technique due to the fact that it better detects larger shifts.

4. EWMA schemes may be applied for monitoring standard deviations in addition to the process mean.

5. EWMA schemes can be used to forecast values of a process mean.

6. The EWMA methodology is not sensitive to normality assumptions.

In real situations, the exact value of the shift size is often unknown and can only be reasonably assumed to vary within a certain range. Such a range of shifts deteriorates the performance of existing control charts. One of the algorithms for determining the maximal shift in normal traffic is described in [17].

The paper describes the process of application of EWMA algorithm for one major user, given as an example. It can be shown that the obtained results are valid for the other analyzed users as well. This research uses samples of authentic network traffic (i.e. traffic intensity in a unit of time). Traffic analysis is realized in the form of statistical calculations on samples which derives from traffic curve. From the appropriate pattern of Internet traffic, 35 samples of local maximums are taken in order to ensure that the statistical analysis is performed on a large sample (number of samples $n > 30$), thus supporting and leading to general conclusions.

The aim of this research is to determine those allowed EWMA values of traffic, that when they are exceeded, it will be considered as the appearance of a statistical anomaly suspected to attack. In this sense, the choice of only local maximums for analysis can be accepted as logical, because the critical point of maximum value of aggregate traffic is in this way also included.

The proposed method of calculating the overall optimal value Λ is applied to traffic patterns, on the basis of which the lower and upper control limits of traffic are determined. For statistical detection of an attack the primary interest is the appearance of a situation in which the upper control limit is exceeded. The overstepping of the lower control limit can be understood as a statistical anomaly, but in the case of this research, it is only related to the local maximum (and not to the aggregate network traffic) and as such does not endanger the security of the computer network in general. Therefore, the situation in which the value of network traffic falls below some lower limit is not considered to be a suspicious event or attack, because the initial presumption of this research is the increase of traffic during an external attack. For the observed pattern of traffic, EWMA values are calculated and if these values are outside of the control limits, that situation is interpreted as a statistical anomaly. Emphasis in this work is placed on determining the occurrence of false alarms, as an important security feature of applied algorithm.

2 Optimized Exponential Smoothing

Calculating the optimal value of parameter λ is based on the study of authentic samples of network traffic. Random variations of network traffic are normal phenomena in the observed sample. In order to decrease or eliminate the influence of individual random variations of network traffic on occurrence of false alarms, the procedure of exponential smoothing is applied, as an aspect of data preprocessing.

For any time period t, the smoothed value S_t is determined by computing:

$$S_t = \lambda y_{t-1} + (1-\lambda)S_{t-1} \quad \text{where} \quad 0 < \lambda \leq 1 \text{ and } \quad t \geq 3 \tag{4}$$

This is the basic equation of exponential smoothing. The formulation here is given by Hunter [2].

This smoothing scheme begins by setting S_2 to y_1, where S_i stands for smoothed observation or EWMA, and y_i stands for the original observation. The subscripts refer to the time periods 1, 2, ..., n. For example, the third period is $S_3 = \lambda y_2 + (1-\lambda)S_2$ and so on. There is no S_1. The optimal value for λ is the value which results in the smallest mean of the squared errors (MSE).

The initial EWMA plays an important role in computing all the subsequent EWMA's. There are several approaches to define this value:

1. Setting S_2 to y_1

2. Setting S_2 to the target of the process

3. Setting S_2 to average of the first four or five observations

It can also be shown that the smaller the value of λ, the more important is the selection of the initial EWMA. The user would be well-advised to try several methods before finalizing the settings.

For different input values of initial parameter S_2, an application in "Matlab" is created which calculates and plots the dependence of SSE and partial value of λ in range of $0 \div 1$, with adjustable step. In addition, the optimal value λ_{opt} is also calculated. For the optimal value, in accordance with the smoothing scheme, that particular value is taken for which the SSE is minimal. The following figure shows an example for calculating the optimal value of the parameter λ for a specific S_2.

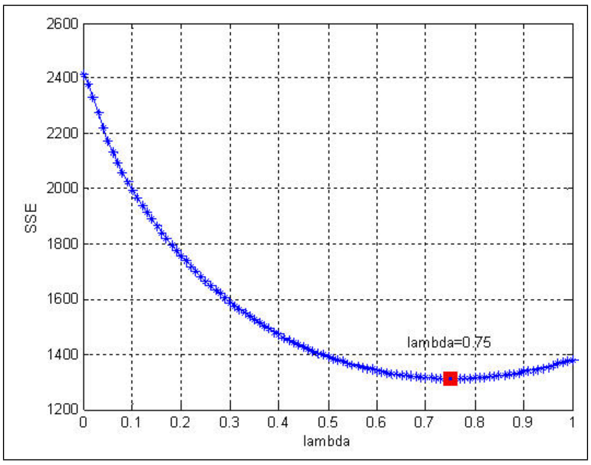


Figure 1: Calculation λ_{opt} (SSE)

Due to the lack of an exact method of calculation in available publications about the determination of the initial S_2 in the procedure of exponential smoothing, the authors of this paper have dealt with researching the link between selection of $S_2 = y_1$ and λ_{opt}, i.e. $S_2(\lambda_{opt})$. In that sense, the range of S_2 is determined using the lowest to the highest sample value during the period of observation. This research was conducted on an authentic sample of network traffic of an Internet service provider and the segment of observation was the range of values of local maximums (in this concrete case from 8 to 34 Mb/s), with a large enough number of values $n = 33 > 30$, taking into account the generality of conclusions. The period of observation was one month. The next table (Table 1) shows the numerical and graphical dependence $S_2(\lambda_{opt})$.

Since a set of different results has been obtained for partial values of λ_{opt}, the authors suggest for the overall optimal parameter λ_{opt} to accept the average of all the partial results (in this particular case it is 0.75).

3 Autocorrelation

Autocorrelation or serial correlation of time series means that the value of the observed variable in a time unit depends on values which appear prior to or later in series. In practical situations, autocorrelation of the first order is usually examined, which may be shown by a simple correlation coefficient or so-called autocorrelation coefficient. Let R_t be the time series data, where $t = 1, 2, \ldots, T$, then the autocorrelation coefficient of the first order is given by:

	S_2	λ_{opt}
1	8	0.72
2	9	0.72
3	10	0.72
4	11	0.72
5	12	0.71
6	13	0.71
7	14	0.72
8	15	0.72
9	16	0.72
10	17	0.72
11	18	0.72
12	18.5	0.73
13	19	0.73
14	19.5	0.73
15	20	0.73
16	20.5	0.73
17	21	0.74
18	21.5	0.74
19	22	0.74
20	22.5	0.75
21	23	0.75
22	23.5	0.75
23	24	0.75
24	25	0.76
25	26	0.77
26	27	0.77
27	28	0.78
28	29	0.79
29	30	0.8
30	31	0.8
31	32	0.81
32	33	0.82
33	34	0.82

Table 1: Calculation of $S_2(\lambda_{opt})$

$$\rho(R) = \frac{\sum_{t=2}^{T} R_t R_{t-1}}{\sqrt{\sum_{t=2}^{T} R_t^2 \sum_{t=2}^{T} R_{t-1}^2}} \qquad \text{-1} \leq \rho \leq 1 \tag{5}$$

One of the standard features of traffic time series is that increasing rates of traffic R_t are not mutually significantly autocorrelated, ie. the value of autocorrelation coefficient is near zero. At the same time, this means that the distribution of positive and negative values of increasing rates is random and that does not follow a specific systematic regularity. Positive autocorrelation implicates that the positive values are followed by mainly positive values and negative values by negative ones and then is $\rho \approx +1$. In the case of negative autocorrelation, there is often a change of sign, i.e. the positive rate in most cases leads to a negative rate and vice versa and then is $\rho \approx -1$. Since there is no typical scheme, on the basis of positive rate in particular time period there is now way of concluding (it can not be concluded) with a significant probability that in the next period either a growth or decline will appear. The same applies to the situation for the negative rate.

Researchers in [5] dealt with the influence of autocorrelated and uncorrelated data on the behavior of intrusion detection algorithm. In their work they came to conclusion that EWMA algorithm for autocorrelated and uncorrelated data works well in the sense of intrusion detection in some information systems. The advantage of EWMA technique for uncorrelated data is that this technique (as opposed to the case of autocorrelated data) can detect not only rapid changes in the intensity of events, but also small changes in the mean value realized through the gradual increase or decrease of the intensity of events. However, in EWMA for uncorrelated data, the initial value of smoothed intensity events is to be reset after intrusion detection, in order to avoid the impact of current values of parameters on future results (*carry-over effect*). In case of EWMA for autocorrelated data this reset is not necessary, because EWMA automatically adjusts the upper and lower control limits. Generally, the smoothing constant should not be too small, so that a short-term trend in the intensity of events in the recent past can be detected.Other publications have also shown the need for taking into account the autocorrelation of input data. As it is emphasized in [18], in the case of dynamic systems the autocorrelation in variables is taking into account incorporating time lags of the time series during the modeling stage.

Samples of network traffic were obtained by network software "MRTG" (Multi Router Traffic Grapher) version 2.10.15. This software generates three types of graphs:

- Daily - with calculation of 5-minute average
- Weekly - with calculation of 30-minute average
- Monthly - with calculation of 2-hour average

The graphs also enable numerical information on the maximum and average traffic for the appropriate period of time.

Daily, weekly and monthly graphs of the first measurement will be used for calculation of initial historical data, while the application of EWMA statistics, with aim of checking the validity of certain parameters, will be realized on daily, weekly and monthly traffic graphs of the second measurement.

For the application of exponential smoothing method to network traffic it is necessary first to determine the historical values: $EWMA_0$ and standard deviation σ_0. For this purpose, it is necessary to collect appropriate traffic samples to perform adequate calculations. This study will use a total of 105 samples of local maximum: 35 samples from the daily traffic graph, 35 samples from the weekly traffic graph and 35 samples from the monthly traffic graph (Table 2).

Time	y_t(daily)	y_t(weekly)	y_t(monthly)
1	12	21	23
2	10.5	22.5	30
3	8.5	23	27
4	10.5	20	27
5	18	20.5	25
6	22	23.5	27
7	25.5	24	22
8	20	21	24
9	33.9	23	23
10	25	25	20
11	24	25.5	24.5
12	26.5	24.5	26.5
13	27.5	22	28
14	23	25.5	27
15	25	27	23
16	24	28	22.5
17	23	27	26.5
18	23	28	31
19	22	25.5	22.5
20	23	30	22.5
21	23	29	27
22	23	26.5	25
23	23	29	26
24	16	26.5	28
25	16	27.5	21
26	9	26	24
27	11.5	25	22
28	8.5	24	22
29	8.5	23.5	22
30	14	22	23
31	23	22.5	27
32	23	24	29
33	20	24	25
34	23	25	25
35	23	23	22

Table 2: Network samples

On the basis of the data presented in the table the following can be calculated: $EWMA_0$ = 23.10 and $\sigma_0 = 4.87$.

In accordance with the method described above and to justify the usage of EWMA statistics, it is important to determine the statistical independence of samples, which will be examined by checking the existence of correlation between data. For this purpose, Pearson's correlation coefficient will be used, which is supplied as a ratio of covariances of two variables and the product of their standard deviations:

$$\rho_{xy} = \frac{Cov(X,Y)}{\sigma_x \sigma_y} \qquad -1 \leq \rho_{xy} \leq 1 \tag{6}$$

Other authors proposed different interpretation ways of correlation coefficient. Cohen [1] noted that all the criteria are based on the greater or lesser extent of arbitrariness and should not be kept too strictly. Yet, one often used interpretation of these coefficients is given below, as described in [16]:

- ρ between 0 and 0.2 - no correlation or is insignificant
- ρ between 0.2 and 0.4 - low correlation
- ρ between 0.4 and 0.6 - moderate correlation
- ρ between 0.6 and 0.8 - significant correlation
- ρ between 0.8 and 1 - high correlation

The value of correlation coefficient ρ_{xy} can be calculated using the statistical function CORREL (array1, array2) in MS Excel. When examining the table above, it is possible to identify three series of data (daily, weekly and monthly) and in this sense three different correlation coefficients can be calculated:

- correlation coefficient for daily - weekly series: $\rho_1 = 0.28 \rightarrow$ low correlation
- correlation coefficient for daily - monthly series: $\rho_2 = 0.04$
- correlation coefficient for weekly - monthly series: $\rho_3 = -0.04$

Beside testing the correlation coefficient within a single measurement, it is important to check the existence of correlation between corresponding periods from different measurements. For that purpose, values of correlation coefficient of two daily (weekly, monthly) intervals are checked and the following results are obtained:

- correlation coefficient for daily - daily series: $\rho_4 = -0.15$
- correlation coefficient for weekly - weekly series: $\rho_5 = 0.11$
- correlation coefficient for monthly - monthly series: $\rho_6 = -0.02$

As all calculated coefficients are with low degree of correlation, or without it, it can be concluded that the used data are statistically independent and that the application of EWMA statistics is justified.

4 Network Practice

Values $EWMA_0$ and σ_0 are calculated for the period of one month. It can be reasonably assumed that these values in another monthly period of observation would be different, having in mind the various unpredictable traffic situations and accidental nature of network traffic as a process. Therefore, in the further phase of research, the extent of change of these values will be studied.

The determination of maximum changes of characteristic traffic values (maximum and average) is based on the analysis of numerical data of several larger Internet users that derives from the popular network software MRTG, which is related to the period of one day, week and month. Without the loss of generality, the graphical presentation of curves from three users is given below, noting that the observed traffic curves of other users do not deviate significantly from the forms shown here.

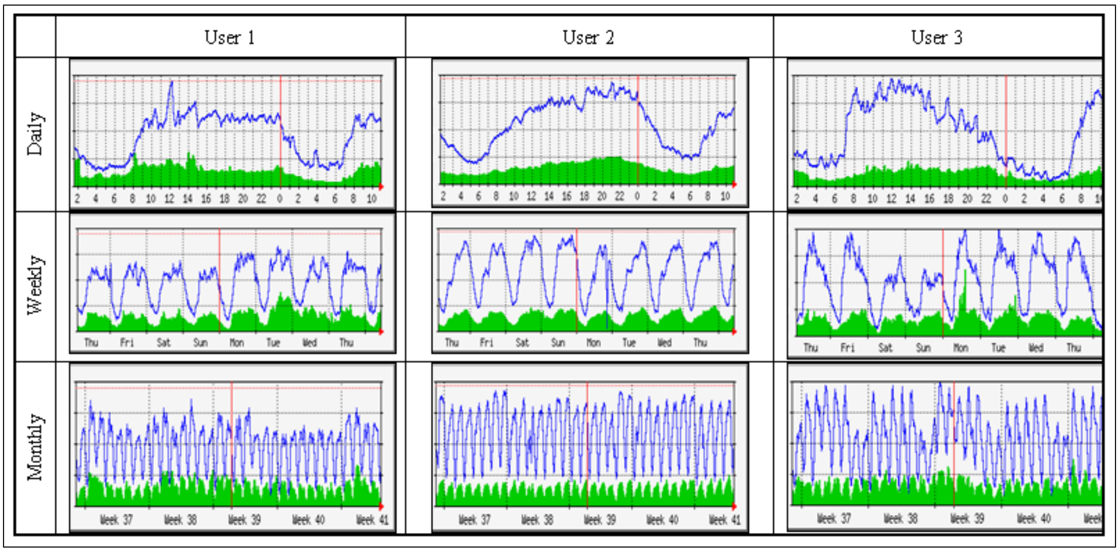


Figure 2: Traffic curves of different users

	Daily1 [Mb/s]	Daily2 [Mb/s]	Diff. [%])	Weekly1 [Mb/s]	Weekly2 [Mb/s]	Diff. [%]	Monthly1 [Mb/s]	Monthly2 [Mb/s]	Diff. [%]
User 1									
Max	33.9	33.1	-2.4	29.7	33.4	12.4	9.7	9.8	1
Average	16.5	19.1	15.8	17.0	21.1	24.1	6.01	6.6	9.8
User 2									
Max	3.94	3.63	-7.8	3.98	3.68	-7.5	48.2	49.2	2
Average	2.35	2.09	-11	2.28	2.09	-8.3	30.9	30	-3
User 3									
Max	9.31	10.0	7.4	9.71	9.99	2.9	9.9	9.7	-2
Average	5.71	6.01	5.2	5.63	6.64	17.9	5.4	4.9	-9.2
User 4									
Max	9.69	9.99	3.1	10.0	9.91	-0.9	10	10	0
Average	4.96	5.14	3.6	5.2	4.94	-5	7.4	7.6	2.7
User 5									
Max	48.2	46.3	-3.9	48.5	45.2	-6.8	1.8	1.8	0
Average	29	24.4	-15.9	30.4	26.4	-13.1	0.14	0.14	0
User 6									
Max	10.1	10.1	0	10.0	10.0	0	3.94	3.66	-7.1
Average	7.78	7.95	2.2	7.43	8.14	9.6	1.9	2.03	6.8
User 7									
Max	3.98	3.97	-0.02	3.94	3.99	1.2	3.9	3.9	0
Average	1.74	1.79	2.9	1.88	1.99	5.9	1.9	2	5.2

Table 3: Differences in characteristic values of traffic

The maximum and average values of traffic are calculated twice in the period of a month. The results are arranged in the following table:

By analysis of the numerical data it can be concluded that the maximum change of mean value p is not greater than 25%. Namely, that is the largest deviation of size in both measurements.

Since the main idea of this research is to find out the maximum and minimum tolerant EWMA values of local maxima, in further calculations the following values will be used (for the accepted p = 0.25):

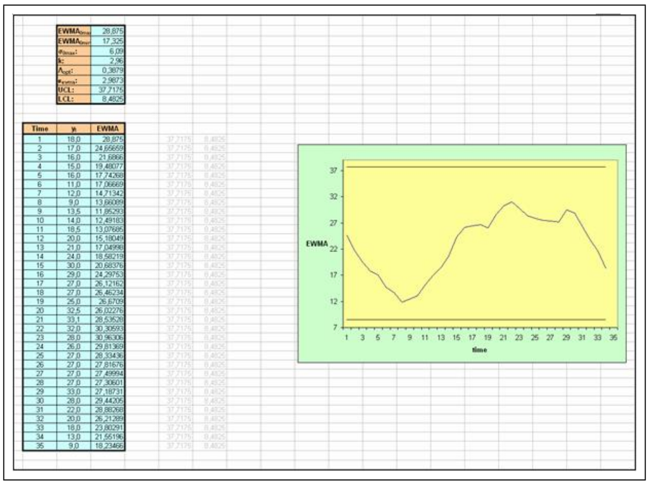
$EWMA_{0max} = (1+p)EWMA_0 \approx 1.25 EWMA_0$

$EWMA_{0min} = (1-p)EWMA_0 \approx 0.75 EWMA_0$

Similarly, $\sigma_{0max} \approx 1.25\sigma_0$ and $\sigma^2_{EWMA} = (\Lambda/(2-\Lambda))\sigma^2_{0max}$.

Previously calculated values of parameters $EWMA_0$ and Λ_{opt}, as well as the values of UCL and LCL from the first measurement will be verified in different situations of daily, weekly and monthly traffic from the second measurement. UCL value is given by $EWMA_{0max} + k\sigma_{EWMA}$, while LCL is $EWMA_{0min} - k\sigma_{EWMA}$. The following results are obtained:

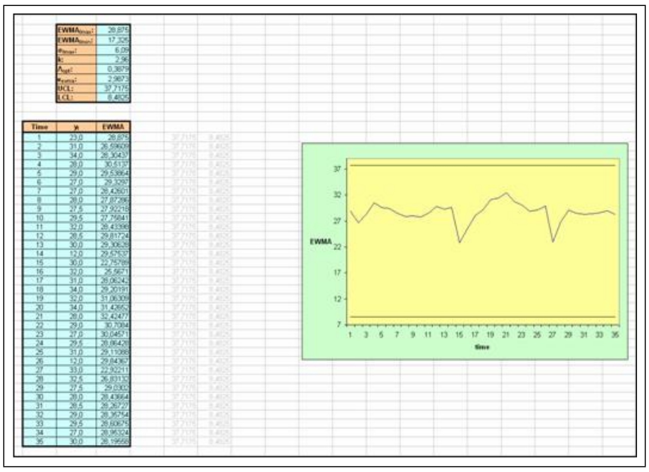
It is important to notice that according to the previous figure in case of appropriately determined EWMA parameters, there is no situation of threshold exceeding, which eliminates the appearance of false alarms.



$EWMA_{0max}$:	28,875
$EWMA_{0min}$:	17,325
σ_{0max}:	6,09
k:	2,96
λ_{opt}:	0,3879
σ_{ewma}:	2,9873
UCL:	37,7175
LCL:	8,4825

Time	y_t	EWMA
1	18,0	28,875
2	17,0	24,65669
3	16,0	21,6866
4	15,0	19,48077
5	16,0	17,74268
6	11,0	17,06669
7	12,0	14,71342
8	9,0	13,66089
9	13,5	11,85293
10	14,0	12,49183
11	18,5	13,07686
12	20,0	15,18049
13	21,0	17,04998
14	24,0	18,58219
15	30,0	20,68376
16	29,0	24,29753
17	27,0	26,12162
18	27,0	26,46234
19	25,0	26,6709
20	32,5	26,02276
21	33,1	28,53528
22	32,0	30,30693
23	28,0	30,96306
24	26,0	29,81369
25	27,0	28,33436
26	27,0	27,81676
27	27,0	27,49994
28	27,0	27,30601
29	33,0	27,18731
30	28,0	29,44205
31	22,0	28,88268
32	20,0	26,21289
33	18,0	23,80291
34	13,0	21,55196
35	9,0	18,23466

Figure 3: Verification (daily traffic, second measurement)



$EWMA_{0max}$:	28,875
$EWMA_{0min}$:	17,325
σ_{0max}:	6,09
k:	2,96
λ_{opt}:	0,3879
σ_{ewma}:	2,9873
UCL:	37,7175
LCL:	8,4825

Time	y_t	EWMA
1	23,0	28,875
2	31,0	26,59609
3	34,0	28,30437
4	28,0	30,51137
5	29,0	29,53864
6	27,0	29,3297
7	27,0	28,42601
8	28,0	27,87286
9	27,5	27,92215
10	29,5	27,75841
11	32,0	28,43396
12	28,5	29,81724
13	30,0	29,30628
14	12,0	29,57537
15	30,0	22,75789
16	32,0	25,5671
17	31,0	28,06242
18	34,0	29,20191
19	32,0	31,06309
20	34,0	31,42652
21	28,0	32,42477
22	29,0	30,7084
23	27,0	30,04571
24	29,5	28,86428
25	31,0	29,11088
26	12,0	29,84367
27	33,0	22,92211
28	32,5	26,83132
29	27,5	29,0302
30	28,0	28,43664
31	28,5	28,26727
32	29,0	28,35754
33	29,5	28,60675
34	27,0	28,95324
35	30,0	28,19658

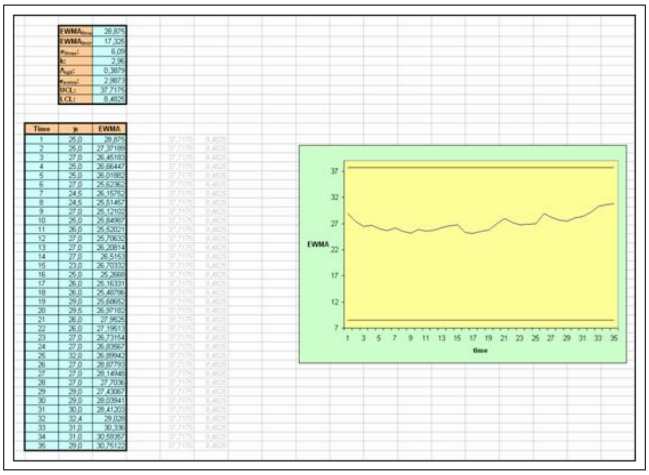
Figure 4: Verification (weekly traffic, second measurement)

5 Conclusions

The aim of this research was to examine the possibility of applying EWMA statistics in intrusion detection in network traffic.

The research has shown that direct application of this algorithm on computer network traffic, as applied in industrial processes, does not provide acceptable results. Namely, often proposed values for exponential smoothing factor in case of network application of the algorithm, may in some circumstances lead to the creation of false alarms, thus endangering the security level of system. Due to the lack of an acceptable precise method for determination of initial value of the coefficient in exponential smoothing procedure in available publications, this research has been directed towards establishing a relation between the choice of initial ratio and optimal value for smoothing. By creating the appropriate application, the practical way was presented for testing the impact of different values of parameters on the level of anomaly detection. This enabled the establishment of graphical presentation of input depending on output sizes, which all contributed to the creation of proposed method for calculating the optimal value of smoothing factor.

Before the start of the implementation of statistical analysis of traffic, the extent of autocorrelation between the used data has to be examined, by calculating the correlation coefficients. One of the impor-



EWMA[illegible]	28,875
EWMA[illegible]	17,325
σ[illegible]:	6,09
k:	2,96
λ_{opt}:	0,3879
σ[illegible]:	2,9873
UCL:	37,7175
LCL:	8,4825

Time	y	EWMA
1	25,0	28,875
2	25,0	27,37188
3	27,0	26,45183
4	25,0	26,66447
5	25,0	26,01882
6	27,0	25,62362
7	24,5	26,15752
8	24,5	25,51457
9	27,0	25,12102
10	25,0	25,84987
11	26,0	25,52021
12	27,0	25,70632
13	27,0	26,20814
14	27,0	26,5153
15	23,0	26,70332
16	25,0	25,2668
17	26,0	25,16331
18	26,0	25,48786
19	29,0	25,68652
20	29,5	26,97182
21	26,0	27,9525
22	26,0	27,19513
23	27,0	26,73154
24	27,0	26,83967
25	32,0	26,89942
26	27,0	28,87793
27	27,0	28,14948
28	27,0	27,7036
29	29,0	27,43067
30	29,0	28,03941
31	30,0	28,41203
32	32,4	29,028
33	31,0	30,336
34	31,0	30,59357
35	29,0	30,75122

Figure 5: Verification (monthly traffic, second measurement)

tant results is that it is shown that analysis of properties of network traffic based on individual patterns of daily traffic only is not recommended, because of the increased level of autocorrelation. For this reason, when calculating the historical parameters, network traffic must be viewed in a wider context of time, taking into account the weekly and monthly periods. Using the network monitoring software, it is also necessary to determine the maximum variations of basic traffic characteristics (average and maximum).

To make this algorithm properly applicable in network environment it is necessary to perform previous processing of historical data, in order to obtain initial values of key parameters.

Based on the proof lent by the obtained results it can be concluded that the choice of EWMA parameters significantly affects the operation of this algorithm in network environment. Therefore, the optimization process of parameters before the application of the algorithm is of particular importance.

Bibliography

[1] J. Cohen, Statistical power analysis for the behavioral sciences (2nd ed.), Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1998.

[2] J.S. Hunter, The exponentially weighted moving average, Journal of Quality Technology 18: 203-210, 1986.

[3] J.M. Lucas, M.S. Saccucci, Exponentially weighted moving average control schemes: Properties and enhancements, Technometrics 32, 1-29., 1990.

[4] S.W. Roberts, Control Chart Tests Based on Geometric Moving Averages, *Technometrics*, 1959.

[5] Ye et al., Computer Intrusion Detection Through EWMA for Autocorrelated and Uncorrelated Data, *IEEE Transactions on Reliability* vol. 52, No. 1, 2003.

[6] G. Fengmin, Deciphering Detection Techniques: Part II Anomaly-Based Intrusion Detection, *White Paper, McAfee Security*, 2003

[7] S. Sorensen, Competitive Overview of Statistical Anomaly Detection, *White Paper, Juniper Networks*, 2004.

[8] V. A. Mahadik, X. Wu and D. S. Reeves, Detection of Denial-of-QoS Attacks Based on χ^2 Statistic And EWMA Control Charts, *http://arqos.csc.ncsu.edu/papers/2002-02-usenixsec-diffservattack.pdf*

[9] A. S. Neubauer, The EWMA Control Chart: Properties and Comparison with other Quality-Control Procedures by Computer Simulation, *Clinical Chemistry, http://www.clinchem.org/cgi/content/full/43/4/594*

[10] D. Seibold, Enterprise Campus Security-Addressing the Imploding Perimeter, *http://www.itsa.ufl.edu/2003/presentations/IntSec.ppt*

[11] A. Vasilios, S. and F. Papagalou, Application of Anomaly Detection Algorithms for Detecting SYN Flooding Attacks, *http://www.ist-scampi.org/publications/papers/siris-globecom2004.pdf*

[12] J. Viinikka and H. Debar, Monitoring IDS Background Noise Using EWMA Control Charts and Alert Information, *http://viinikka.info/ViiDeb2004.pdf*

[13] Y. Zhao, F. Tsung and Z. Wang, Dual CUSUM Control Schemes for Detecting a Range of Mean Shifts, *IEEE Transactions, http://qlab.ieem.ust.hk/qlab/download/papers/paper%2035.pdf*, 2005

[14] Engineering Statistics Handbook-EWMA Control Charts, *http://www.itl.nist.gov/div898/handbook/pmc/section3/pmc324.htm*

[15] Engineering Statistics Handbook–Single Exponential Smoothing, *http://www.itl.nist.gov/div898/handbook/pmc/section4/pmc431.htm*

[16] Savannah State University, Office of Institutional Research & Planning, *http://irp.savstate.edu/irp/glossary/correlation.html*

[17] P. Čisar, S. Maravić Čisar, A first derivate based algorithm for anomaly detection, *International journal of computers, communications & control, 3(S):238-242, 2008*

[18] J. Mina, C. Verde, Fault Detection for Large Scale Systems Using Dynamic Principal Components Analysis with Adaptation, *International journal of computers, communications & control, 2(2):185-194, 2007..*

Petar Čisar was born on September 08, 1965. He graduated at the Faculty of Electrical Engineering in Belgrade. Master's study completed in information engineering at the Faculty of Economics in Subotica. Currently works on PhD thesis. The spheres of his interest are mobile technologies, as well as the development of security methods in network environments.

Saša Bošnjak was born on December 31, 1961. He is a associate professor of Computer Science at the Faculty of Economics Subotica. He holds a range of courses in information engineering. His research interests are database, software development, computer networks, reuse methodology, ebusiness and internet technology. He earned a PhD degree in Information System Faculty of Economics Subotica in 1995.

Sanja Maravić Čisar was born on August 03, 1970. She graduated at the Faculty of Electrical Engineering in Belgrade. Master's study completed at the Technical Faculty in Zrenjanin. She works as lecturer at Subotica Tech at following courses: Visual programming, Object-oriented based programming, JAVA and Multimedia systems. Currently works on PhD thesis.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 2, pp. 171-178

Environmental Policy and Science Management: Using a Scientometric-specific GIS for E-learning Purposes

N.D. Hasanagas, A.D. Styliadis, E.I. Papadopoulou

Nikolaos D. Hasanagas, Ahanasios D. Styliadis
Kavala Institute of Technology, Department of Landscape Architecture
City of Drama, Greece, GR 66100
Email: styliadis@ath.forthnet.gr, nikolaos.hasanagas@gmail.com

Eleni I. Papadopoulou
Aristotle University of Thessaloniki, Faculty of Agricultural Science
Department of Agricultural Economics, City of Thessaloniki, GR 54124
E-mail epapa@agro.auth.gr

Abstract:
Who is the "good scientist" in rural-environmental policy? This is not so self-evident as in the case of private high-tech industry. Developing e-learning system in environmental science management is a challenging task in the area of forest and general rural development policy. Who determines the most "important" scientific information and who controls it? There are algorithms for measuring centrality in information networks. The concepts of closeness and betweenness centrality are used as basic metadata for categorizing the communication type in the rural-environmental policy networks. This paper discusses the development of a GIS-based model which includes region-based scientometrics, regarding policy field communication.

Keywords: region-based scientometrics, GIS e-Learning, forest policy, integrated rural development policy, environmental policy, complete network analysis, policy making.

1 Introduction

It is evident that in high-tech policy sector, "good scientist" is the private industry, since impressive products like mobile phones are usually produced by the industry and not by universities or public agencies. However, this is not the case in rural-environmental policy, where the main issues are about "multiple-use forestry", "nature protection", "integrated rural development", "sustainability" etc, as in these abstract issues, success normally means to persuade about who the "trouble-maker" and who the "rescuer" is and not to find concrete solutions. Moreover, despite the common belief that universities or research institutes are the most influential suppliers and controllers of "scientific" information, this is also not necessarily the rule in rural-environmental policy.

This paper presents the possibility of developing a pattern of scientometric-specific GIS -based model, which is going to be applicable in e-learning for various target groups (e.g. students who are specialized in forest policy and rural policy analysis, lobbyists, policy makers). In the context of this paper, "scientometric" differs from the classical measurements of citation rates or implementation of academic bibliography. It is defined as the management of information which is regarded as "scientific" by lobbyists, public administrators and academics in networks of forest policy issues which are used as case studies in this paper.

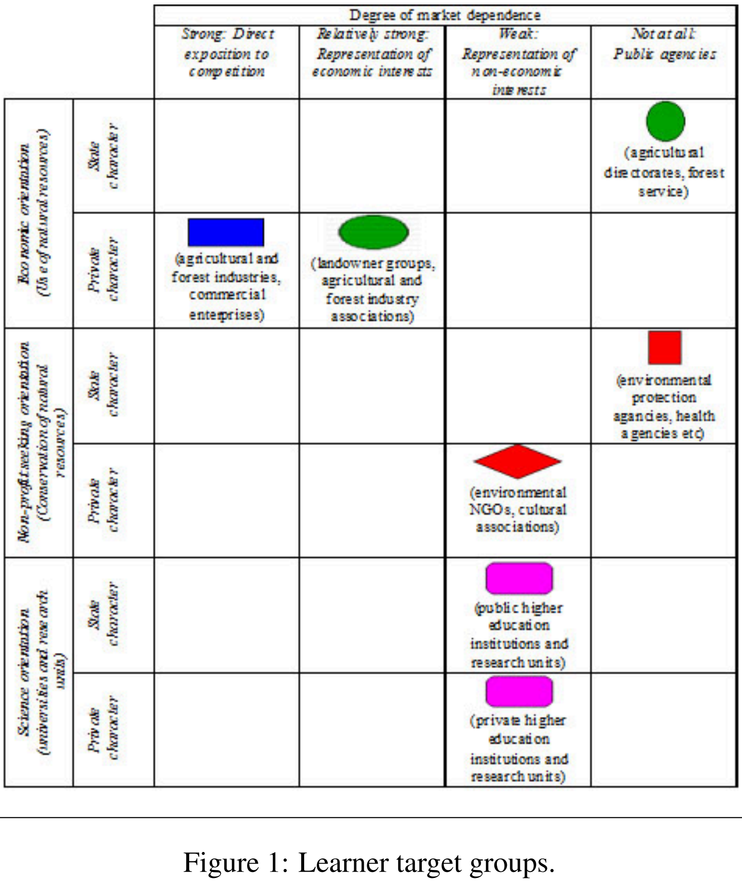
The complete network analysis conducted here by VISONE software showed that the "scientific" information, which is regarded as most "important" in a network, is often supplied by practitioners, such as environmental and forest agencies, landowner organizations etc, and not necessarily by actors officially entitled as "scientific", such us universities or research centers. This implies that the "scientific" is rather a feeling induced by factors different from the official orientation of an actor. The analysis pointed out that forest policy, environmental issues and rural development are inseparable policy fields and strongly integrated with each other.

The actors who produce "scientific" information, which is considered to be "important", can disseminate the information which is favorable for them, in order to support their socio-political positions. Thus, a "scientifically important" actor may be an informal but influential decision-maker. Additionally, the network will collapse, if an actor who participates in many information paths and thus controls the flow of information, suddenly abandons the network. In this case, the actor is merely a crucial "postman" but not necessarily a "decision-maker". In policy planning, implementation and evaluation, it is useful to distinguish the "decision-maker" from the "postman" in order to focus the lobbying activity on the former rather than on the latter. This is especially useful in integrated rural development, incl. forest policy, as there are numerous complex, informal and different procedures among the countries and unclear indicators (RUDI research project findings). Therefore, there is a point in developing an adaptive GIS e-learning system on region-specific information networks in order to make lobbyists, public administrators and students more familiar with science management in policy arena [14].

2 Selecting the GIS Learning Object and the learner target group

Adaptive learning systems are supposed to require numerous and elaborated rules [5, 6, 18]. However, the existence of tested algorithms which detect the position of an actor in the formal or informal hierarchy of information in a network simplifies the rules and metadata requirements, even in complex social systems [6, 8, 11, 16, 19]. Two algorithms, "closeness centrality" (cc) and "centrality of betweenness" (cb), can be used for classifying the actors. The only question is which type of actors interests each learner group [2, 11, 17, 6, 9]. We can distinguish two aggregated and seven specific learner groups:

A) Aggregated learner groups: 1) State actor employees (civil servants of agencies), 2) Private actor employees (lobbyists, managers, experts and employees of interest groups or enterprises)



		Degree of market dependence			
		Strong: Direct exposition to competition	*Relatively strong: Representation of economic interests*	*Weak: Representation of non-economic interests*	*Not at all: Public agencies*
Economic orientation (Use of natural resources)	*State character*				● (agricultural directorates, forest service)
	Private character	■ (agricultural and forest industries, commercial enterprises)	⬬ (landowner groups, agricultural and forest industry associations)		
Non-profit seeking orientation (Conservation of natural resources)	*State character*				🟥 (environmental protection agancies, health agencies etc)
	Private character			◆ (environmental NGOs, cultural associations)	
Science orientation (universities and research units)	*State character*			▬ (public higher education institutions and research units)	
	Private character			▬ (private higher education institutions and research units)	

Figure 1: Learner target groups.

B) Specific learner target groups (Figure 1): These emerge by the combination of three dimensions [12, 13, 15]: 1) the legal character of the actor (public or private), 2) the orientation of the actor (economic or non-profit seeking regarding the management of natural resources, or science-orientation), and 3) the degree of dependence on the liberal market (i. individual enterprises which are directly exposed to the competition, ii. economic interest groups of enterprises which only represent their interests in the policy network, iii. non-profit seeking interest groups such as cultural association for the renovation of villages, forest recreation, and environmental NGOs, which only depend on national and EU funds, or private universities and research centres which can be flexible to the market needs, and iv. state actors which are totally independent from the market, such as agencies, public universities and research units) (RUDI findings).

The GIS learning objects (Figure 2) are the seven actor types described in Figure 1 (“Archimedes” research project findings), in each category of information management function (cc and cb). An enterprise characterized by high cc (strong "decision-maker") in a certain region-specific policy network is a case of learning object, while the same enterprise characterized by a low cb (weak "postman") is a different case of learning object.

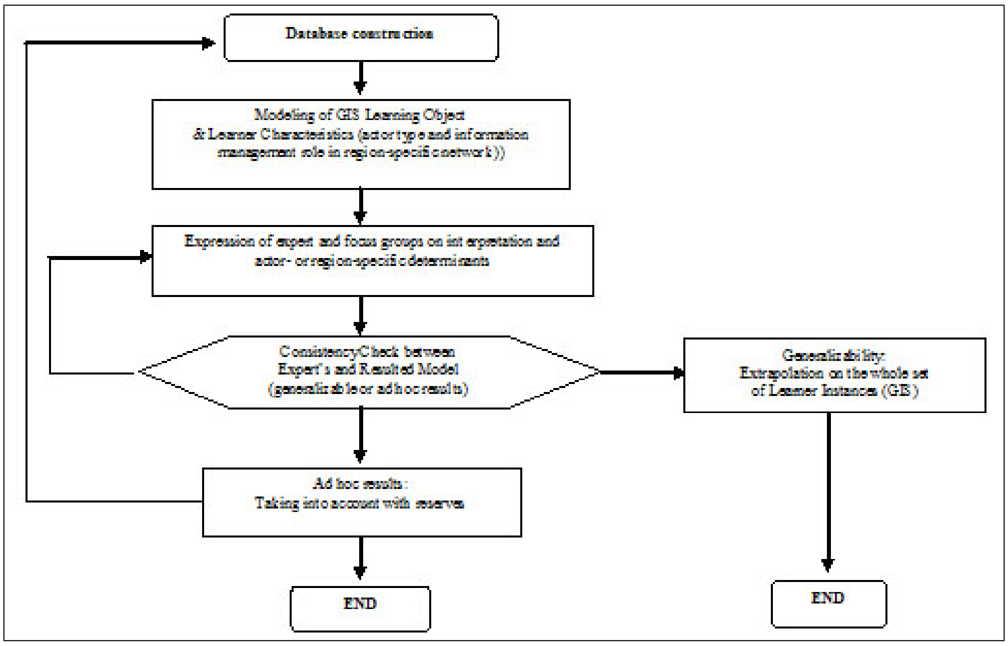

Database construction
Modeling of GIS Learning Object
& Learner Characteristics (actor type and information management role in region-specific network))
Expression of expert and focus groups on interpretation and actor- or region-specific determinants
ConsistencyCheck between Expert's and Resulted Model (generalizable or ad hoc results)
Generalizability: Extrapolation on the whole set of Learner Instances (GB)
Ad hoc results: Taking into account with reserves
END
END

Figure 2: Processing Model Framework.

Each learner can thus examine whether the actor he/she belongs to is a strong "decision-maker" or "postman" in comparison to the other actors, so as to plan his/her lobbying strategy more effectively. In this way, instead of a standardized instructional design, the learner- practitioner (or policy analysis student) can be flexibly oriented by using the cc and cb as indicators.

3 Visualizing "decision-makers" and "postmen"

Various socio-informatics software products have been used for measuring and visualizing the invisible structures of policy networks which very often include informal relationships [1, 3]. The INSNA web-site ("International Network for Social Network Analysis"- www.insna.org) offers a remarkable variety of such software. However, the implementation and outputs of this software in e-learning through a region-specific GIS Learning Management System is still an open challenge, where the close cooperation of a multi-disciplinary group of social scientists [13, 15], software engineers, forest policy analysts, economists and practitioners such as lobbyists, foresters, agrarians etc is necessary. The informal relations are sometimes much more decisive for the policy output than the formal ones [3, 4, 7, 14]. A usual relation type in such region-specific networks is the exchange of information [8, 9]. The VISONE software is appropriate for depicting and analyzing both formal and informal information exchange. The algorithms of cc and cb are used for this purpose. The cc is an indicator of importance of a policy actor, while the cb indicates the capacity of controlling the paths of information exchange and thereby the dissemination of "scientific" information [13].

Cc measures the distance d (i.e. the shortest number of links) between two actors. If i.e. the Royal Scottish Forestry Society gives information to the Friends of the Loch Lomond and the latter to the National Trust of Scotland (and there is no direct information link from the Royal Scottish Forestry Society to the National Trust of Scotland), then the distance (d) between the Royal Scottish Forestry Society and the National Trust of Scotland is d=2 (links). The sum of all distances from an actor i to any other actor is the closeness of the actor i and then the closeness centrality of i is defined as its inverse closeness [8, 9]:

$$Cc_{(i)} = \lfloor \sum_j d(j,i) \rfloor^{-1} \tag{1}$$

The fewer links are needed to connect i to any other actor, the higher its cc is. If an actor possesses information that is regarded as crucial by the other actors, then one can expect this actor to have a high cc. For this reason, cc is an indicator of the information importance.

Cb quantifies the information control potential of an actor i and is defined as the sum of the ratios of shortest paths between other actors that the actor i sits on:

$$Cb_{(i)} = \sum \frac{|P_i(i,j)|}{|P(i,j)|} \tag{2}$$

where P(i,j) and Pi (i,j) are the sets of all shortest paths between i and j, and those shortest paths passing through i, respectively. Thus, an actor with high cb plays the role of the "go-between" for many other actors in term of shortest paths and, in this way, controls the dissemination of information. If the actor is also aware of its capability, it can influence the decision-making processes.

Cb points out the most crucial "postman" of information in a network, while cc shows who is the most important "sender". The "sender" is an information imposer and thereby a "decision-maker" who can directly influence the policy-making. In everyday life, the "sender" (e.g. a bank which makes a decision) may often be clearly differentiated from the person who brings the letter from the bank to the interested client. However, in politics, it is often unclear who the "sender" and who the "postman" is, as these roles are informal. Using the two indicators, cb and cc, this difference can be detected.

Three examples of region-specific networks of "scientific" information flow are presented in Figure 3. These are networks investigating issues like UK1: Management of Loch Lomond and Trossachs National Park, UK2: Scottish Forestry Strategy, and Greece: The amendment of the forest-related Article 24 of the National Constitution Law. The actors are Landowners' Associations, Forest Services, Agricultural Directorates, Cultural Associations (related to nature aesthetics and cultural landscapes), Angling and Hunting Associations, Environmental NGOs, Technical Chambers, Tourist Enterprises, Forest and Agricultural Industries, Municipalities etc (Figure 1 can be used as a legend to understand the nodes in the depicted networks). Number of actors, number of links and the network density (%) appear in each table cell of Figure 3.

In the first row, the whole region-specific networks appear (two from Scotland and one from Greece). The networks of "scientific" information shaped according to the cc and cb of the actors are presented in the second and third row respectively.

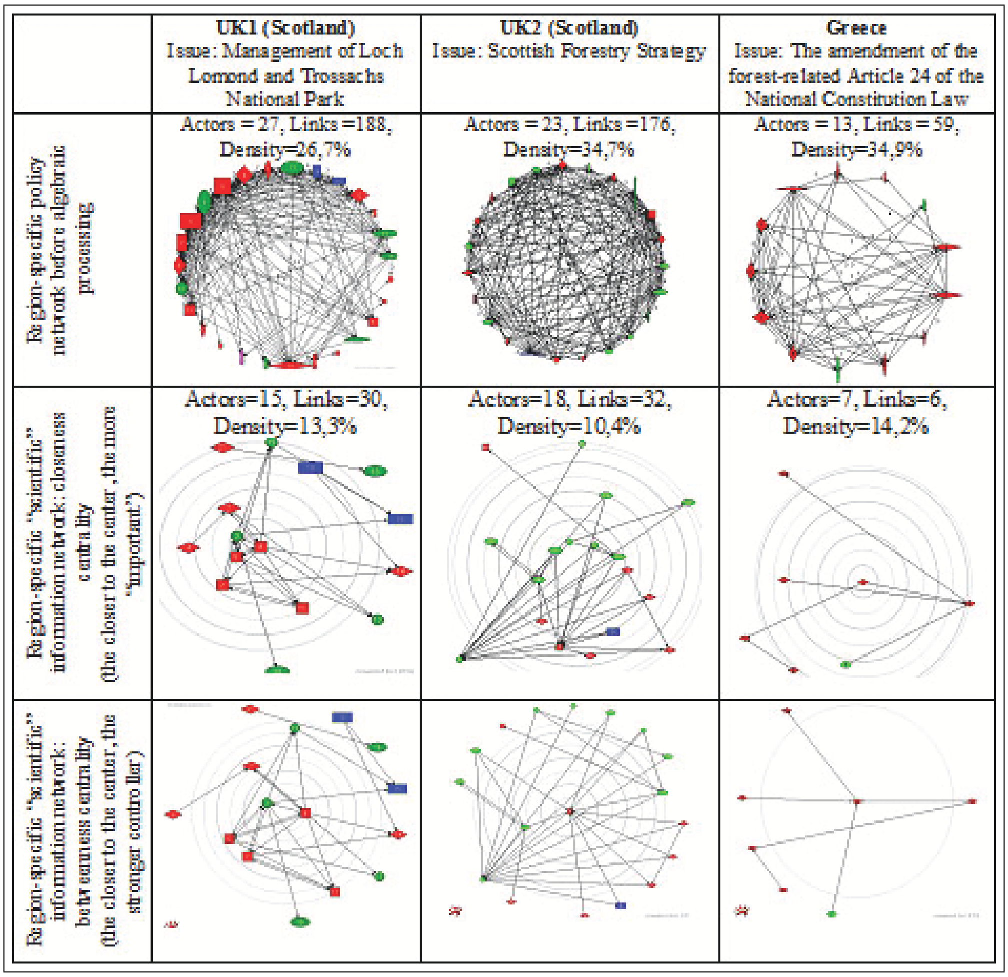
The difference of the density degree (%) between whole contact networks (first row) and this of "scientific" information networks is remarkable: The density of "scientific" information is drastically lower than the "general contact" density, though issues like forest-environment and integrated rural development policy are supposed to be multi-disciplinary areas. Not only the "science"-interested actors are relatively few but the lower degree of density also shows that the "scientific" communication is much weaker than the total activities (e.g. lobbying and institutional pressure, exchange of material support etc) (RUDI findings).

The analysis suggests that in some networks, economic or profit-seeking, private or state actors appear to be the most "important" actors in the "scientific" arena, while universities or research institutes are quite low in importance. This is particularly true for a science-oriented actor (a university) in the UK1 issue. This university is totally excluded from the "scientific" information network. Thus, it is quite subjective which information is regarded as "scientific" (RUDI findings).

Moreover, by comparing the cc with the cb network of the UK2 issue, it becomes obvious that the most influential "decision-maker" is not necessarily the best "postman": A Forest Service possesses the highest cc, while an environmental protection agency possesses the highest cb.

In Figure 4, a comparative view of region-specific scientometric data is provided: Each learner target group can focus its research work on the respective learning object (actor type and position in information hierarchy).

The learners can examine and interpreting the quantitative results: e.g. why the most important "scientist" and "science" controller is the state, why the science-oriented actors are of disputable importance and much weaker in intermediating "science" etc. The e-learning system functions as an "adaptor" which connects the academic knowledge with the practical and "local knowledge". The synergy between field expert experience and academic knowledge can be thereby optimized. Possibilities of coalition building or cross-sectoral over-bridging through science management [12, 14] as well as chances for effective coordination through information control [7, 10, 15] can be discussed and analyzed by all learners. Such an e-learning system could be useful in research projects which aim at the better understanding of rural-environmental policy design, delivery, monitoring and evaluation



	UK1 (Scotland) Issue: Management of Loch Lomond and Trossachs National Park	**UK2 (Scotland)** Issue: Scottish Forestry Strategy	**Greece** Issue: The amendment of the forest-related Article 24 of the National Constitution Law
Region-specific policy network before algebraic processing	Actors = 27, Links =188, Density=26,7%	Actors = 23, Links =176, Density=34,7%	Actors = 13, Links = 59, Density=34,9%
Region-specific "scientific" information network: closeness centrality (the closer to the center, the more "important")	Actors=15, Links=30, Density=13,3%	Actors=18, Links=32, Density=10,4%	Actors=7, Links=6, Density=14,2%
Region-specific "scientific" information network: betweenness centrality (the closer to the center, the stronger controller)			

Figure 3: Region-specific networks of "scientific" information flow.

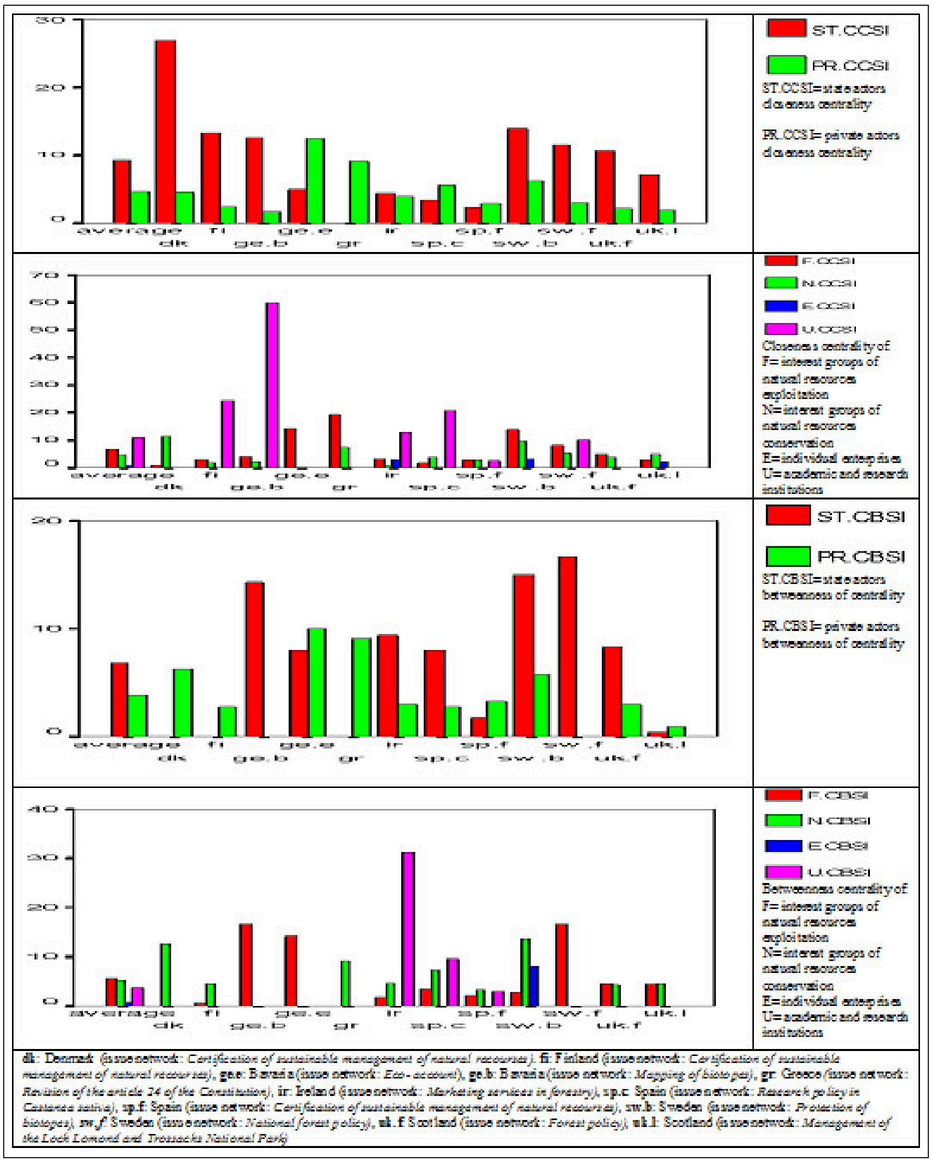
("Archimedes" and RUDI findings). In particular, the implementation of forest policy as a holistic "scientific" toolbox for socio-economic planning, financing and development of forestry, evaluation of the protective and economic role of forest, prediction of supply and demand of forest products and forest administration is enabled.

4 Summary and Conclusions

There are different science management patterns among region-specific policy networks. Science-oriented actors are not necessary the leading ones in the information networks which are considered as "scientific" ones by their participants. A scientometric-specific GIS based on complete network analysis helps all learners understand that forest policy, integrated rural development and environmental policy constitute a multifaceted and cohesive politico-administrative system, where the potential influence regarding information use and distribution becomes quite complex and makes the policy output unpredictable. By the presented GIS e-learning system all learner target groups become more familiar with a policy design which takes into account the needs not only of the agricultural population but of a whole integrated rural territory, which is developed in the framework of a new type of urban-countryside relationship (RUDI findings). The quantified learning objects of the integrated rural policy arena should be further interpreted through a participative and qualitative approach. Thereby, the socio-economic and ecological driving forces which determine the quantitative results will be better understood by the learner target groups, independently of any political interests or personal views.

Acknowledgements

The research initiative proposed by this paper has been supported by the EU-funded research project "Archimedes"



dk: Denmark (issue network: *Certification of sustainable management of natural recourses*). **fi**: Finland (issue network: *Certification of sustainable management of natural recourses*), **ge.e**: Bavaria (issue network: *Eco- account*), **ge.b**: Bavaria (issue network: *Mapping of biotopes*), **gr**: Greece (issue network: *Revision of the article 24 of the Constitution*), **ir**: Ireland (issue network: *Marketing services in forestry*), **sp.c**: Spain (issue network: *Research policy in Castanea sativa*), **sp.f**: Spain (issue network: *Certification of sustainable management of natural recourses*), **sw.b**: Sweden (issue network: *Protection of biotopes*), **sw.f**: Sweden (issue network: *National forest policy*), **uk.f**: Scotland (issue network: *Forest policy*), **uk.l**: Scotland (issue network: *Management of the Lock Lomond and Trossachs National Park*)

Figure 4: Scientometric- embedded GIS.

(Department of Landscape Architecture, Kavala Institute of Technology, Drama, Greece), by the EU-funded research project "RUDI: Rural Development Impacts- Assessing the impact of Rural Development policies, incl. LEADER" (Department of Agricultural Economics, Faculty of Agricultural Science, Aristotle University of Thessaloniki, Greece), and by the Institute of Forest Policy and Nature Conservation of Goettingen University (Germany).

Bibliography

[1] K.A. Bakar, B. S. Doherty, *Evaluation of the Recorded State Mechanism for Protecting Agent Integrity Against Malicious Hosts*. Int. J. of Computers, Communications & Control, 3(1), 60-68, 2008.

[2] A.D. Styliadis, I.I. Akbaylar, D.A. Papadopoulou, N.D. Hasanagas, S.A. Roussa, L.A. Sexidis, *Metadata-based heritage sites modeling with e-learning functionality*. Journal of Cultural Heritage, 10, 296-312, 2009.

[3] M. A. Rajan, M. Girish Chandra, L.C. Reddy, P. Hiremath, *Concepts of Graph Theory Relevant to Ad-hoc Networks*. Int. J. of Computers, Communications & Control, 3(S):465-469, 2008.

[4] P. Borne, M. Benrejeb, *On the Representation and the Stability Study of Large Scale Systems*. Int. J. of Computers, Communications & Control, 3(S):55-66, 2008.

[5] A.D. Styliadis, D.G. Konstantinidou, K.A. Tyxola, *eCAD System Design - Applications in Architecture*. Int. J. of Computers, Communications & Control, 3(2), 204-214, 2008.

[6] A.D. Styliadis, *Digital documentation of historical buildings with 3-d modeling functionality*. Automation in Construction, 16, 498-510, 2007.

[7] N.D. Hasanagas, *Lobbying management: Principles and methods of a more effective practice in the environmental labyrinth of the European Union*. Scottish Forestry. The journal of the Royal Scottish Forestry Society. 57(1), 28-32, 2003.

[8] M. Baur, M. Benkert, U. Brandes, S. Cornelsen, M. Gaertler, B. Koepf, J. Lerner, D. Wagner, *Visone - Software for Visual Social Network Analysis*. Porc. 9th Intl. Symp. Graph Drawing, Lecture Notes in Computer Science 2265, 463-464, Springer, 2002.

[9] A.D. Styliadis, *Historical photography-based computer-aided architectural design: Demolished buildings information modeling with reverse engineering functionality*. Automation in Construction, 18, 51-69, 2008.

[10] N. Zarour, S. Bouzidi, *Coalition Formation for Cooperative Information Agent-Based Systems*. Int. J. of Computers, Communications & Control, 1(3), pp. 85-92, 2006.

[11] A. Andreatos, *Virtual Communities and their Importance for Informal Learning*. Int. J. of Computers, Communications & Control, 2(1), 39-47, 2007.

[12] M. Krott, N.D. Hasanagas, *Measuring bridges between sectors: Causative evaluation of cross- sectorality*. Forest Policy and Economics, 8, 555-563, 2006.

[13] KM. Krott, *Catalyst for innovation in European forest policy sciences. Evaluation of the EFI Research Program 3: policy analysis*. Forest Policy and Economics, 5(2), 123-134, 2003.

[14] S. Shortall, *Are rural development programmes socially inclusive? Social inclusion, civic engagement, participation, and social capital: Exploring the differences*. Journal of Rural Studies, 24(4), 450-457, 2008.

[15] D.J.A. Douglas, *The restructuring of local government in rural regions: A rural development perspective*. Journal of Rural Studies, 21(2), 231-246, 2005.

[16] A.D. Styliadis, I.D. Karamitsos, D.I. Zachariou, *Personalized e-Learning Implementation - The GIS Case*. International Journal of Computers, Communications & Control, 1(1), 59-67, 2006

[17] A.D. Styliadis, *E-Learning Documentation of Historical Living Systems with 3-D Modeling Functionality*. INFORMATICA, 18(3), 419-446, 2007.

[18] A.D. Styliadis, P.G. Patias, N.C. Zestas, *3-D Computer Modeling with Intra-Component, Geometric, Quality and Topological Constraints*. INFORMATICA, 14(3), 375-392, 2003.

[19] A.D. Styliadis, M.Gr. Vassilakopoulos, *A spatio-temporal geometry-based model for digital documentation of historical living systems*. Information & Management, 42, 349-359, 2005.

Nikolaos D. Hasanagas Born in 1974. Assistant Professor in environment-related subjects at the Kavala Institute of Technology, Drama, Greece. BSc and MSc eq. in Environmental Sciences (Aristotle Univ. of Thessaloniki, Greece), BA and MA eq. in Social Sciences, PhD in Environmental Policy Analysis (Goettingen Univ., Germany).

Athanasios D. Styliadis Born in 1956. Professor of digital architecture and design computing at the Department of Landscape Architecture at the Kavala Institute of Technology, Drama, Greece. Diploma in Surveying Engineering, MSc in Computer Science (Dundee Univ., Scotland), PhD in CAAD and GIS (Aristotle Univ. of Thessaloniki, Greece).

Eleni I. Papadopoulou Born in 1957. Assistant Professor in Rural Policy at the Faculty of Agricultural Science at the Aristotle University of Thessaloniki, Greece. BSc in Agriculture Engineering, MSc in Agricultural Economics (Univ. of Reading, UK), PhD (Aristotle University of Thessaloniki).

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 2, pp. 179-192

Evolved Fuzzy Control System for a Steam Generator

D. Hossu, I. Făgărăşan, A. Hossu, S.S. Iliescu

Daniela Hossu, Ioana Făgărăşan, Andrei Hossu, Sergiu St. Iliescu
University Politehnica of Bucharest
Automatic Control and Industrial Informatics
No.313 Splaiul Independentei, District 6, Code 060042, Bucharest, Romania
E-mail: andone_dana@yahoo.com, dana@aii.pub.ro, ioana@shiva.pub.ro

Abstract:
Poor control of steam generator water level is the main cause of unexpected shutdowns in nuclear power plants. Particularly at low powers, it is a difficult task due to shrink and swell phenomena and flow measurement errors. In addition, the steam generator is a highly complex, nonlinear and time-varying system and its parameters vary with operating conditions. Therefore, there is a need to systematically investigate the problem of controlling the water level in the steam generator in order to prevent such costly reactor shutdowns. The objective of this paper is to design, evaluate and implement a water level controller for steam generators based on a fuzzy model predictive control approach. An original concept of modular evolved control system, seamless and with gradual integration into the existent control system is proposed as base of implementation of the presented system.

Keywords: evolved control, nonlinear system, fuzzy model, predictive control.

1 Introduction

Poor control of the steam generator water level in the secondary circuit of a nuclear power plant can lead to frequent reactor shutdowns. Such shutdowns are caused by violation of safety limits on the water level and are common at low operating power where the plant exhibits strong non-minimum phase characteristics.

Difficulties on designing a steam generator (SG) level controller arise from the following factors:

- reverse dynamics, or non-minimum phase behavior due to swell and shrink effects,
ticularly at low power,
- changes in plant dynamics with operating power,
- dynamics uncertainties,
- corrupted feed-water flow measurement signal with biased noises.

Particularly it is difficult to control automatically a steam generator water level during transient period or at low power less than 15% of full power because of its dynamic characteristics.

Various approaches have been reported in the literature: an adaptive PID level controller using a linear parameter varying model to describe the process dynamics over the entire operating power range [12]; LQG controllers with "gain-scheduling" to cover the entire operating range [15]; a hybrid fuzzy-PI adaptive control of drum level, a model predictive controller to identify the operating point at each sampling time and use the plant model corresponding to this operating point as the prediction model [14].

A nonlinear physical model with a complexity that is suitable for model-based control has been presented by Aström and Bell [5]. The model describes the behavior of the system over a wide operating range. A model of the steam generator water level process in the form of a transfer function, determined based on first-principles analysis and expert experience has been presented in [30]. A detailed nonlinear model based on the lumped parameter approach for system modules, developed for a typical PWR power plant has been presented in [1]. In the paper trained neural networks are used to predict certain system parameters of the plant for a number of different power demand histories. Paper [17] presents a self organizing fuzzy logic controller for the water level control of a steam generator.

With the advent of the current generation of high-speed computers, more advanced control strategies not limited to PI/PID, can be applied [11], [18], [20], [23]. Model predictive control (MPC) is one such controller design technique, which has gained wide acceptance in process control applications. Model predictive control has three basic steps: output prediction, control calculation and closing the feedback loop [6], [7], [16]. In this paper, we apply MPC techniques to develop a framework for systematically addressing the various issues in the SG level control problem. The Takagi-Sugeno fuzzy model representation often provides efficient and computationally attractive solutions to a wide range of modeling problems capable to approximate nonlinear dynamics, multiple operating modes and significant parameter and structure variations [22], [24]. This paper deals with Takagi-Sugeno (T-S) fuzzy models because this type of model have a good capability for prediction and can be easily used to design model-based predictive controllers for nonlinear systems [9].

The paper includes simulations of typical operating transients in the SG.

A new concept of modular advanced control system designed for a seamless and gradual integration into the target systems is presented. The system is designed in such a way to improve the quality of monitoring and control of the whole system. The project targets the large scale distributed advanced control systems with optimum granularity architecture.

2 Fuzzy Model

Fuzzy models have become one of the most well established approaches to non-linear system modeling since they are universal approximations which can deal with both quantitative and qualitative (linguistic) forms of information [8], [9], [21], [26], [27], [28], [29]. These models can broadly be divided into three classes: Linguistic Models - so-called Mamdani-type Models, Fuzzy Relational Models, and Takagi-Sugeno (TS) Models [14]. Both linguistic and fuzzy relational models are linguistically interpretable and can incorporate prior qualitative knowledge provided by experts [25]. However, when linguistic knowledge extraction is not the main purpose of modeling, like in many dynamic system identification and control problems, the use of TS models becomes particularly adequate since they are able to accurately represent a wide class of nonlinear systems using a relatively small number of parameters. In a nutshell, TS models perform an interpolation of local models, usually linear or affine in their arguments, by means of a fuzzy inference mechanism. Their functional rule base structure is well-known to be intrinsically favorable for control applications.

The TS model representation often provides efficient and computationally attractive solutions to a wide range of modeling problems capable to approximate nonlinear dynamics, multiple operating modes and significant parameter and structure variations. The ability of such model to capture the dynamics of a large class of nonlinear plants has been investigated extensively in the literature [13].

In this paper, we apply MPC techniques to develop a framework for systematically addressing the various issues in the SG level control problem. This paper deals with Takagi-Sugeno (T-S) fuzzy models because of their capability to approximate a large class of static and dynamic nonlinear systems. In T-S modeling methodology, a nonlinear system is divided into a number of nearly linear subsystems. A quasi-linear empirical is developed by means of fuzzy logic for each subsystem. The whole process behavior is characterized by a weighted sum of the outputs from all quasi-linear fuzzy implication. The methodology facilitates the development of a nonlinear model that is essentially a collection of a number of quasi-linear models regulated by fuzzy logic. It also provides an opportunity to simplify the design of model predictive control.

The system is divided into a number of linear or nearly linear subsystems. By Takagi-Sugeno's modeling methodology a fuzzy quasi-linear model has to be developed for each subsystem. In such a model, the cause-effect relationship between control *u* and output *y* at the sampling time *n* is established in a discrete time representation. Each fuzzy implication is generated based on a system *step response* [3, 4, 10].

$$\begin{aligned} &IF\ y(n)\ is\ A^i_0,\ y(n-1)\ is\ A^i_1,\ \ldots,\ y(n-m+1)\ is\ A^i_{m-1},\\ &and\ u(n)\ is\ B^i_0,\ u(n-1)\ is\ B^i_1,\ \ldots,\ u(n-l+1)\ is\ B^i_{l-1} \qquad (1)\\ &THEN\ y^i(n+1) = y(n) + \sum_{j=1}^{T} h^i_j\,\Delta u(n+1-j) \end{aligned}$$

where:

A^i_j	fuzzy set corresponding to output *y(n-j)* in the i^{th} fuzzy implication
B^i_j	fuzzy set corresponding to input *u(n-j)* in the i^{th} fuzzy implication
h^i_j	impulse response coefficient in the i^{th} fuzzy implication
T	model horizon
$\triangle u(n)$	difference between *u(n)* and *u(n-1)*

A complete fuzzy model for the system consists of *p* fuzzy implications. The system output *y(n+1)* is inferred as a weighted average value of the outputs estimated by all fuzzy implications

$$y(n+1) = \frac{\sum_{j=1}^{p} \mu^j y^j(n+1)}{\sum_{j=1}^{p} \mu^j} \qquad (2)$$

where

$$\mu^j = \wedge_i A^j_i \wedge_k B^j_k \qquad (3)$$

considering

$$\omega^j = \frac{\mu^j}{\sum_{j=1}^{p} \mu^j} \qquad (4)$$

then

$$y(n+1) = \sum_{j=1}^{p} \omega^j\, y^j(n+1) \qquad (5)$$

3 Fuzzy Model Predictive Control

3.1 Problem formulation

The design goal of a fuzzy model predictive control is to minimize the predictive error between an output and a given reference trajectory in the next N_y steps through the selection of N_u step optimal control policies.

The optimization problem can be formulated as:

$$min_{\Delta u(n),\, \Delta u(n+1),\, \ldots,\, \Delta u(n+N_u)}\ J(n) \qquad (6)$$

$$J(n) = \sum_{i=1}^{N_y} \mu_i(\hat{y}(n+i) - y^r(n+i))^2 + \sum_{i=1}^{N_u} v_i \Delta u(n+i)^2 \qquad (7)$$

where:

μ_i and v_i are the weighting factors for the prediction error and control energy;
$\hat{y}(n+1)$ i^{th} step output prediction;
$y^r(n+1)$ i^{th} step reference trajectory;
$\Delta u(n+i)$ i^{th} step control action.

The weighted sum of the local control policies gives the overall control policy:

$$\Delta u(n+i) = \sum_{j=1}^{p} \omega^j \Delta u^j(n+i) \tag{8}$$

Substituting (2) and (8) into (7) yields (9)

$$J(n) = \sum_{i=1}^{N_y} \mu_i \left(\sum_{j=1}^{p} \left(\omega^j \left(\hat{y}^j(n+i) - y^r(n+i) \right) \right) \right)^2 + \sum_{i=0}^{N_u-1} v_i \left(\sum_{j=1}^{p} \omega^j \Delta u^j(n+i) \right)^2 \tag{9}$$

To simplify the computation, an alternative objective function is proposed as a satisfactory approximation of (9) [10].

$$\widetilde{J}(n) = \sum_{j=1}^{p} \left(\left(\omega^j\right)^2 \left(\sum_{i=1}^{N_y} \mu_i \left(\hat{y}^j(n+i) - y^r(n+i) \right)^2 + \sum_{i=0}^{N_u-1} v_i \, \Delta u^j(n+i)^2 \right) \right) \tag{10}$$

The optimization problem can be defined as:

$$min_{\Delta u(n),\, \Delta u(n+1),\, \ldots,\, \Delta u(n+N_u-1)} \widetilde{J}(n) = min_{\Delta u(n),\, \Delta u(n+1),\, \ldots,\, \Delta u(n+N_u-1)} \sum_{j=1}^{p} \left(\omega^j\right)^2 \widetilde{J^j}(n) \tag{11}$$

where

$$\widetilde{J^j}(n) = \sum_{i=1}^{N_y} \mu_i \left(\hat{y}^j(n+i) - y^r(n+i) \right)^2 + \sum_{i=0}^{N_u-1} v_i \left(\Delta u^j(n+i) \right)^2 \tag{12}$$

Using the alternative objective function (12), we can derive a controller by a hierarchical control design approach.

3.2 Controller design

1. *Lower Layer Design*: For the j^{th} subsystem, the optimization problem is defined as follows:

$$min_{\Delta u(n),\, \Delta u(n+1),\, \ldots,\, \Delta u(n+N_u-1)} \widetilde{J^j}(n) \tag{13}$$

subject to:

$$R_j : \begin{cases} IF\ y(n+k-1)\ is\ A_0^j,\ \ldots,\ y(n+k-m)\ is\ A_{m-1}^j \\ THEN\ y^j(n+k) = y^j(n+k-1) + \sum_{j=1}^{T} h_i^j \Delta u(n+k-i) + \varepsilon^j(n+k-i) \end{cases} \tag{14}$$

Where $\varepsilon^j(n+k-1)$ serves for system coordination and it is determined at the upper layer.

2. *Upper Layer Design*: The upper layer coordination targets the identification of globally optimal control policies through coordinating $\varepsilon^j(n+k-1)$ for each local subsystem.

3. *System Coordination*: From the lower layer, the local information of output and control is transmitted to the upper layer. At the upper layer, the error variables are evaluated as: $\varepsilon^j(n+k-1) = y(n+k-1) - y^j(n+k-1)$. These values will be compared with those for the same error variables calculated in the last iteration. If

$$\sum_{j=1}^{p}\sum_{k=1}^{N_y}\left|e^j(n+k-1) - \varepsilon^j(n+k-1)\right| > \beta$$

then the control policies are not optimal and need to be modified at the lower layer; else an optimal control action is found.

The whole design is decomposed into the derivation of p local controllers. The subsystems regulated by those local controllers will be coordinated to derive a globally optimal control policy. The objective function defined in (12) can be rewritten in a matrix form:

$$\widetilde{J^j}(n) = \left(\hat{Y}^j_+(n) - Y^r(n)\right)^T W^j_1 \left(\hat{Y}^j_+(n) - Y^r(n)\right)_+ + \left(\Delta U^j_+(n)\right)^T W^j_2 \left(\Delta U^j_+(n)\right) \tag{15}$$

where:

$$\hat{Y}^j_+(n) = \left(\hat{y}^j(n+1)\hat{y}^j(n+2)\ldots\hat{y}^j(n+N_y)\right)^T \tag{16}$$

$$Y^r(n) = (y^r(n+1)y^r(n+2)\ldots y^r(n+N_y))^T \tag{17}$$

$$\Delta U^j_+(n) = \left(\Delta u^j(n)\Delta u^j(n+1)\ldots\Delta u^j(n+N_u-1)\right)^T \tag{18}$$

$$W^j_1 = diag\left\{\mu^j_1,\, \mu^j_2,\, \ldots,\, \mu^j_{N_y}\right\} \tag{19}$$

$$W^j_2 = diag\left\{v^j_1,\, v^j_2,\, \ldots,\, v^j_{N_u}\right\} \tag{20}$$

The N_y - step prediction of the output by the j^{th} FI can be rewritten as follows:

$$\hat{Y}^j_+(n) = A^j\Delta U^j_+(n) + Y(n) + P^j(n) + E^j_+(n) \tag{21}$$

where:

$$A^j = \begin{bmatrix} a^j_1 & 0 & 0 & \ldots & 0 \\ a^j_2 & a^j_1 & 0 & \ldots & 0 \\ a^j_3 & a^j_2 & a^j_1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a^j_{N_y} & a^j_{N_y-1} & a^j_{N_y-2} & \ldots & a^j_{N_y-N_u+1} \end{bmatrix} \tag{22}$$

$$a^j_i = \sum_{k=1}^{i} h^j_k \tag{23}$$

$$Y(n) = (y(n)\; y(n)\; \ldots y(n)\;)^T \tag{24}$$

$$P^j(n) = \left(P^j_1(n)\; P^j_2(n)\; \ldots P^j_{N_y}(n)\;\right)^T \tag{25}$$

$$E^j_+(n) = \left(0\sum_{k=1}^{2}\varepsilon^j(n+k-1)\;\cdots\sum_{k=1}^{N_y}\varepsilon^j(n+k-1)\right)^T \tag{26}$$

$$P^j_i(n) = \sum_{k=1}^{i}\sum_{l=k+1}^{T} h^j_l\Delta u(n+k-l) \tag{27}$$

The resulting control policy for the j^{th} subsystem can be derived as:

$$\widetilde{J^j}(n) = \left(\Delta U_+^j(n)\right)^T \left(A^{j^T} W_1^j A^j + W_2^j\right) \Delta U_+^j(n) +$$

$$+ \left(\Delta U_+^j(n)\right)^T A^{j^T} W_1^j Z^j(n) + \left(Z^j(n)\right)^T W_1^j A^j \Delta U_+^j(n) + \left(Z^j(n)\right)^T W_1^j Z^j(n) \tag{28}$$

where:

$$Z^j(n) = Y(n) - Y^r(n) + P^j(n) + E_+^j(n) \tag{29}$$

Minimizing (26) yields:

$$\frac{\delta \widetilde{J^j}(n)}{\delta \Delta U_+^j(n)} = 2(A^{j^T} W_1^j A^j + W_2^j)\Delta U_+^j(n) + 2A^{j^T} W_1^j Z^j(n) = 0 \tag{30}$$

The control law by the j^{th} FI can be identified as:

$$\left(\Delta U_+^j(n)\right)^* = -K^j Z^j(n) \tag{31}$$

where K^j is:

$$K^j = \left(A^{j^T} W_1^j A^j + W_2^j\right)^{-1} A^{j^T} W_1^j \tag{32}$$

The optimal local control policies at the lower layer are identified through optimization, the optimal global control policies can be accordingly derived at the upper layer.

$$\Delta U_+(n) = (\Delta u(n) \Delta u(n+1) \ldots \Delta u(n+N_u-1))^T \tag{33}$$

3.3 Parameter Tuning

In controller design, the difficulty encountered is how to quickly minimize the upper bound of the objective function so that the control actions can force a process to track a specified trajectory as close as possible. There has no rigorous solution to the selection of optimal control horizon *(N_u)* and prediction horizon *(N_y)*. The model horizon is selected so that $T\Delta t \geq$ open loop settling time.

The ranges of weighting factors W_1^j and W_2^j can be very wide, the importance is their relative magnitudes. The following three-step procedure to tune the weighting factors is proposed:

(a) Select a value for W_1 and assign it to all local controllers. Determine W_2^j independently for each local controller in order to minimize the objective function for that subsystem

(b) Identify the largest W_2 and assign it to all subsystems.

(c) Examine the system's closed-loop dynamic performance. If not satisfied, then reduce the value of W_2 gradually until the desirable dynamic performance is identified.

3.4 Simulations

Process Modeling

The main problem in setting up a signal flow diagram for a level controlled system in a SG can be found in the inhomogeneous contents of the SG. The filling consists of water at boiling temperature, pervaded by steam bubbles. Since the volume fraction of the steam bubbles is quite considerable, the mean specific weight of the contents is very strongly dependent on the proportion of steam. This, of

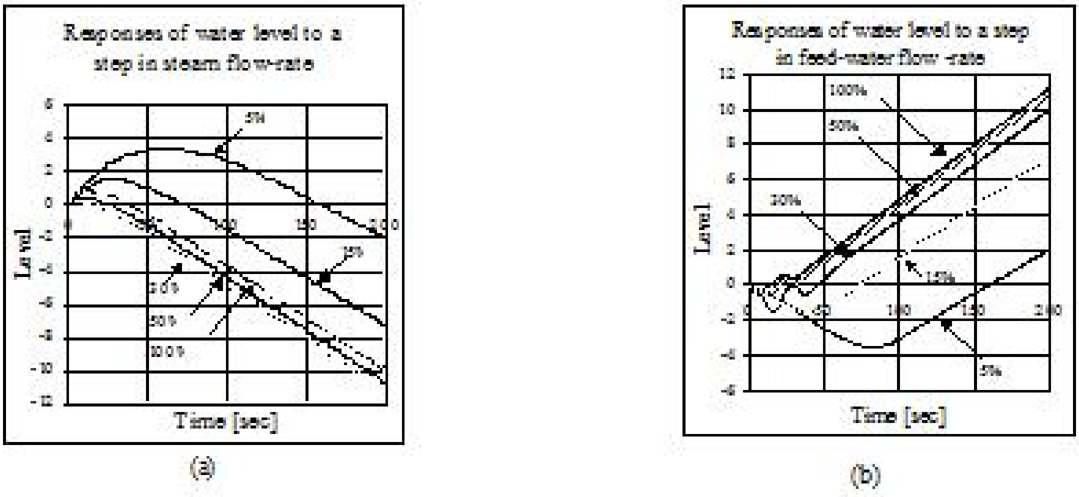


Figure 1: Responses of water level at different operating power (indicated by %) to (a) a step in steam flow-rate. (b) a step in feed-water flow-rate.

course, means that the steam content also strongly influences the level in the SG. The steam content itself depends, in turn, on the load factor, on the changes in feed-water flow, and on feed-water temperature.

The presence of steam below the liquid level in the SG causes the *shrink-and-swell* phenomenon that in spite of an increased supply of water, the water level initially falls. Fig. 1 shows responses of the water level to steps in feed-water and steam flow-rates at different operating powers. For generating the responses, it was used the power dependent linear parameter varying model identified by Irving [12]. At low loads the non-minimum phase behavior is much more pronounced.

The changing process dynamics and the inverse response behavior significantly complicate the design of an effective water level control system. A solution to this problem is to design local linear controllers at different points in the operating regime and then applies gain-scheduling techniques to schedule these controllers to obtain a globally applicable controller.

Consider a step in feed-water flow rate at 5% operating power because of the strong inverse response. For this system, a fuzzy convolution model consisting of four fuzzy implications is developed as follows:

For j=1 to 4
R^j: if $y_{D_{al}}$ (n) is A^j
then $y^j_{D_{al}}(n+1) = y^j_{D_{al}}(n) + \sum_{i=1}^{200} h^j_{D_{al}} u(n+1-i)$

In order to define the fuzzy sets we propose the following strategy: $D_{al_1} = min(y^1_{D_{al}}) \cdot K_{D_{al_1}}$, $D_{al_2} = min(y^2_{D_{al}}) \cdot K_{D_{al_2}}$, $D_{al_3} = min(y^3_{D_{al}}) \cdot K_{D_{al_3}}$, $D_{al_4} = min(y^4_{D_{al}}) \cdot K_{D_{al_4}}$ where $K_{D_{al_1}} = 0.2$, $K_{D_{al_2}} = 0.9$, $K_{D_{al_3}} = 0.9$, $K_{D_{al_4}} = 0.2$. are selected in order to obtain a characteristic as close as possible to the open loop response of water level at 5% operating power to a step in feed-water flow-rate.

Fig. 2 shows the impulse response coefficients for $y^1_{D_{al}}$, $y^2_{D_{al}}$, $y^3_{D_{al}}$, $y^4_{D_{al}}$ subsystems and Fig. 3 shows the definition of fuzzy sets A^1, A^2, A^3 and A^4. Consider a step in steam flow rate at 5% operating power. For this system, a fuzzy convolution model consisting of fourfuzzy implications is developed as follows:

For j=1 to 4
R^j: if y_{D_0} (n) is A^j
then $y^j_{D_0}(n+1) = y^j_{D_0}(n) + \sum_{i=1}^{200} h^j_{D_0} u(n+1-i)$

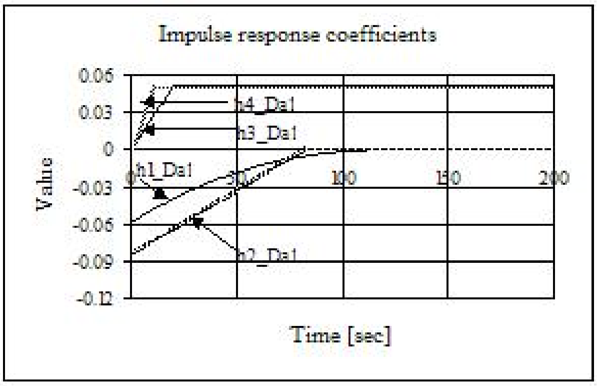


Figure 2: The impulse response coefficients for $y^1_{D_{al}}$, $y^2_{D_{al}}$, $y^3_{D_{al}}$, $y^4_{D_{al}}$ subsystems.

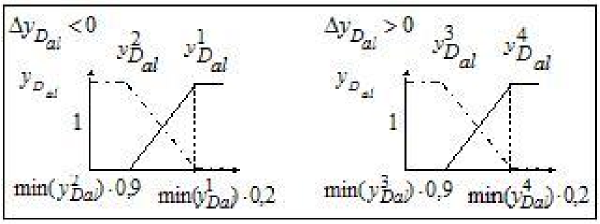


Figure 3: Definition of fuzzy sets A^1, A^2, A^3 *and* A^4 for FI R^1, R^2 ,R^3 *and* R^4 respectively.

In order to define the fuzzy sets we propose the following strategy: $D_{0_1} = max(y^1_{D_0}) \cdot K_{D_{0_1}}$,$D_{0_2} = max(y^2_{D_0}) \cdot K_{D_{0_2}}, D_{0_3} = max(y^3_{D_0}) \cdot K_{D_{0_3}}, D_{0_4} = max(y^4_{D_0}) \cdot K_{D_{0_4}}$ where $K_{D_{0_1}} = 0.4$, $K_{D_{0_2}} = 0.9$, $K_{D_{0_3}} = 0.9$, $K_{D_{0_4}} = 0.6$ are selected in order to obtain a characteristic as close as possible to the open loop response of water level at 5% operating power to a step in steam flow-rate.

Fig. 4 shows the impulse response coefficients for $y^1_{D_0}$, $y^2_{D_0}$, $y^3_{D_0}$, $y^4_{D_0}$ subsystems, Fig. 5 shows the definition of fuzzy sets A^1, A^2, A^3 and A^4.

Controller Design

The goal in this paper is to study the use of the feed-water flow-rate as a manipulated variable to maintain the SG water level within allowable limits, in the face of the changing steam demand resulting from a change in the electrical power demand.

Frequent reactor shutdowns are caused by violation of safety limits on the water level and are

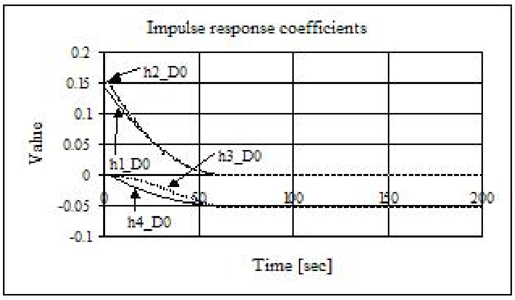


Figure 4: The impulse response coefficients for $y^1_{D_0}$, $y^2_{D_0}$, $y^3_{D_0}$, $y^4_{D_0}$ subsystems.

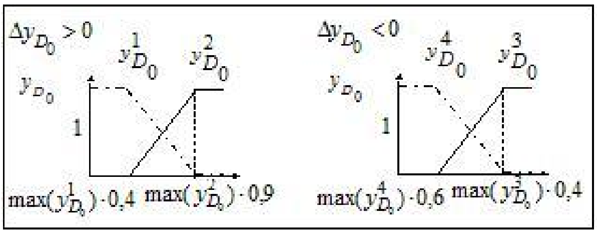

$\Delta y_{D_0} > 0$
$y^1_{D_0}$
$y^2_{D_0}$
y_{D_0}
1
$\max(y^1_{D_0})\cdot 0.4$
$\max(y^2_{D_0})\cdot 0.9$
$\Delta y_{D_0} < 0$
$y^4_{D_0}$
$y^3_{D_0}$
y_{D_0}
1
$\max(y^4_{D_0})\cdot 0.6$
$\max(y^3_{D_0})\cdot 0.4$

Figure 5: Definition of fuzzy sets A^1, A^2, A^3 *and* A^4 for FI R^1, R^2 ,R^3 *and* R^4 respectively.

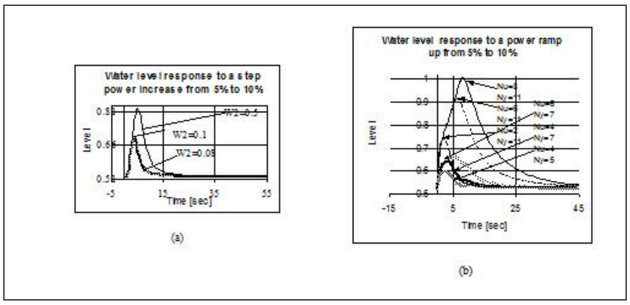


Figure 6: Normalized water level setpoint.

common at low operating power where the plant exhibits strong non-minimum phase characteristics.Fig. 6 shows the normalized water level setpoint and safety limits.

The goal of fuzzy model predictive control is to minimize the predictive error between an output and a given reference trajectory in the next N_y (prediction horizon) steps through the selection of N_u (control horizon) step optimal control policies.

Using the alternative objective function (12), we can design a controller by a hierarchical control design approach.

The whole design is decomposed into the derivation of 4 local controllers. The subsystems regulated by those local controllers will be coordinated to derive a globally optimal control policy. The objective function defined in (12) can be rewritten in a matrix form as follows:

$$\widetilde{J}^j(n) = \left(\hat{Y}^j_+(n) - Y^r(n)\right)^T W^j_1 \left(\hat{Y}^j_+(n) - Y^r(n)\right)_+ + \left(\Delta U^j_+(n)\right)^T W^j_2 \left(\Delta U^j_+(n)\right) \tag{34}$$

where:

$$\hat{Y}^j_+(n) = \left(\hat{y}^j_{D_{al}}(n+1)\hat{y}^j_{D_{al}}(n+2)\ldots\hat{y}^j_{D_{al}}(n+N_y)\right)^T \tag{35}$$

$$Y^r(n) = (y^r(n+1)y^r(n+2)\ldots y^r(n+N_y))^T \tag{36}$$

$$y^r(n) = y^{ref}(n) - y_{D_0}(n) \tag{37}$$

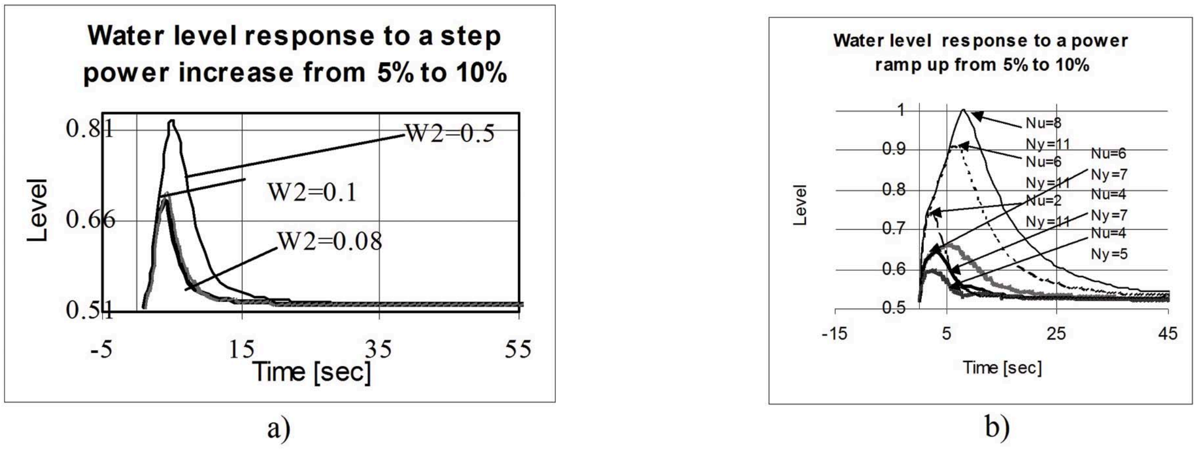
$$\Delta U^j_+(n) = \left(\Delta u^j(n)\Delta u^j(n+1)\ldots\Delta u^j(n+N_u-1)\right)^T \tag{38}$$

$$W^j_1 = diag\left\{\mu^j_1,\, \mu^j_2,\, \ldots,\, \mu^j_{N_y}\right\} \tag{39}$$

$$W^j_2 = diag\left\{v^j_1,\, v^j_2,\, \ldots,\, v^j_{N_u}\right\} \tag{40}$$

The simulations are organized around two different power transients:

- a step-up in power from 5% to 10% (Fig. 7(a));
- a ramp-up in power from 5% to 10% (Fig. 7(b)).



a) b)

Figure 7: Water level response to (a) a step power increase from 5% to 10% (Nu=2, Ny=3, W1=1). (b) a power ramp up from 5% to 10% (W2=0.1, W1=1).

The model horizon is $T = 200$. Increasing N_y results in a more conservative control action that has a stabilizing effect but also increases the computational effort.

The computational effort increases as N_u is increased. A small value of N_u leads to a robust controller.

We can see that the performance is not strongly affected by the presence of the feed-water inverse response, only a slight oscillation is visible in the water level response. All local controllers are used all the time. This means that there is no switch from one local controller to the other in operation. The system output is infered as a weighted average value of the outputs of all subsystems. On the other hand, the overall control policy to the process under control is the weighted sum of all local control policies. This kind of design not only eliminates the controller switch problem and thus possible system instability, but also provides a much more smooth control performance in process operation. The performance is not strongly affected by the presence of the feed-water inverse response, only a slight oscillation is visible in the water level response. The FMPC responses are very satisfactory and not very sensitive to changes in tuning parameters.

3.5 Evolved Controller Client/Server Architecture

An original concept of modular *evolved* control system, seamless and with gradual integration into the *primary* control system is proposed. The target systems are the *large scale distributed control systems* with *optimum granularity architecture*. The aim of the application is to integrate the concepts of *evolved control algorithms, portability of software modules, real time characteristics of the application*. We propose an approach of a *seamless integration* of the evolved control modules into an existing control system. The first part of the life cycle phases of the new control system, from conception to validation stage, the new control system lives hiding in the shadow of the control system it will replace, and after validation the old system will be replaced by the new one. The identification, modeling, control and validation stages of the life cycle of the system, will be done *on-line* (the new system uses a real image of the I/O process data), without affecting the existing control system.

Because of high level of *interconnectivity between system components*, it is necessary to provide the *highest independence between communication modules on one-hand and the control modules on the other hand*. In order to obtain high ability of integration, the communication modules have to cover the widest possible area of industrial communication interfaces and protocols.

One item of the application is to offer a *unified API of extended generality and extendibility in order* *to unify access and information retrieval from various wireless and wired technology and communication interfaces* (RS 232, RS 485, fieldbus: Profibus / Interbus, Ethernet IP, TCP/IP, etc). Applications could properly adapt to changes in the network connections. The design and implementation of a solution *to hide the embedded communication network problems from the application system programmers* is included.

A software package for *evolved control* includes a method based on *fuzzy model predictive control*. By using the basic concept of decomposition-coordination in a large-scale system theory, the *fuzzy model predictive controller* design can be accomplished through a *two-layer iterative design process*. The design is decomposed into the derivation of local controllers. The subsystems regulated by those local controllers will be coordinated to derive a *globally optimal control policy*.

One of the main objectives of the application is to supply *an integrated solution of systems, which should support all the phases of the life cycle: modeling, simulation, development and implementation*. For *parameter tuning, for validation and also for embedding a large number of industrial communication protocols, multi-disciplinary simulation environments are developed which generate instruments for control, I/O data consistency check, and defect detection*. In the end, *real-time advanced control applications* are developed, with *seamless* and *gradual integration into the existing distributed control system*.

In order to provide the real-time characteristic, we choose a multitasking environment for the application (WINDOWS Operating System). From structural point of view we propose a Client / Server architecture for fuzzy Controller (FC) [2]:

Client - is a Windows application representing the implementation of the graphical user interface (GUI). The Client enables the operator to control the system in two modes: manual/automatic, to monitor the system response, etc. The Client has also the ability to connect and communicate with the Server application.

Server is an ActiveX EXE application containing the implementation of the FC kernel. The Server includes a collection of objects, these objects cover the tasks of both data processing and the communication between dedicated applications for input and output data.

The Client application will have a thread pool architecture. The Server application will have a real multithreading architecture (each active object having assigned its own execution thread). The Server have also a multi-layer structure: at the higher level are implemented upper FC and the communication classes (using different transmission mechanisms DDE, OPC, HLI, ActiveX, Winsocket, Pipes), at the lower level are implemented the controllers for the subsystems corresponding to the low level FC. The Server's application as real multithreading architecture, provides the FC Kernel the real-time response characteristic, required for the industrial process control.

4 Conclusions

Control of SG water level strongly affects nuclear power plant availability. The control task is difficult for a number of reasons, the most important among them being the nonlinear plant dynamics and the non-minimum phase plant characteristics. There has been a special interest in this problem during low power transients because of the dominant reverse thermal dynamic effects known as shrink and swell.

The SG level control problem was viewed as a single input/single output control problem with the feed-water as the manipulated variable, the level as the controlled variable and the turbine steam demand as disturbance. The process non-linearity was addressed by scheduling the model (and the controller) with the power level. The SG system is modeled by Takagi-Sugeno's fuzzy modeling methodology, where the system output is estimated based on gradient. The complex shrink and swell phenomena associated with the SG water level are well captured by the model. The predictive controller based on fuzzy model is designed in a hierarchical control design.

An original concept of modular evolved control system, seamless and gradual integration into the existing distributed control system is proposed in the paper. A unified API of extended generality and extendibility in order to unify access and information retrieval from various wireless and wired technology and communication interfaces is developed in order to ensure independence between communication and control modules of the designed systems.A Client / Server architecture for evolved controller that runs on the Windows environment, with real-time characteristics is proposed.

Bibliography

[1] Akkurt, Colak U., PWR system simulation and parameter estimation with neural networks, *Annals of Nuclear Energy*, Vol. 29, pp. 2087-2103, 2002.

[2] Andone D. ,Dobrescu R., Hossu A., Dobrescu M., Application of fuzzy model predictive control to a drum boiler, *ICAE - Integrated Computer-Aided Engineering*, IOSPress, 13(4):347-361, 2006.

[3] Andone D., Hossu A., Predictive Control Based on Fuzzy Model for Steam Generator, *2004 IEEE International Conference on Fuzzy Systems, Proc. FUZZ-IEEE 2004*, vol. 3, IEEE Catalog Number 04CH37542, ISBN 0-7803-8353-2, ISNN 1098-7584, Budapest, Hungary; pp. 1245-1250; July 25-29, 2004.

[4] Andone D., Fagarasan I., Dobrescu M., Advanced Control of a Steam Generator, *The 3rd International IEEE Conference Intelligent Systems, September 04-06, 2006 3rd International IEEE Conference Intelligent Systems*, Vol.s 1 and 2, London, United Kingdom, pp. 338-343, 2006.

[5] Aström K., Bell R., Drum-boiler dynamics, *Automatica* 3, pp. 363-378, 2000.

[6] Camacho E., Bordons C., *Model Predictive Control*, Springer-Verlag, London, 2004.

[7] Demircioglu H., Karasu E., Generalized Predictive Control - A Practical Application and Comparation of Discrete and Continuous-Time Versions, *IEEE Control Systems*, Oct 2000, 20(5):36-44, 2000.

[8] Dubois D., Prade H., *Fuzzy Sets and Systems: Theory and Applications*, Academic Press, Inc., Orlando, FL, 1997.

[9] Espinosa J., Hadjili M. L., Wertz V.,and Vanderwalle J., Predictive control using fuzzy models. Comparative study, *European Control Conference*, Karlsruhe, Germany, Sept. 1999.

[10] Huang Y., Lou Helen H., Gong J.P., Edgar Th. F., Fuzzy Model Predictive Control, *IEEE Trans. On Fuzzy Systems*, 8(6):665-668, 2000.

[11] Hirota K., *Industrial Applications of Fuzzy Technology*, Springer-Verlag, New York, 1993

[12] Irving E., Miossec C., and Tassart J., Toward efficient full automatic operation of the PWR steam generator with water level adaptive control, *Proc. Int. Conf. Boiler Dynamics Contr. Nuclear Power Stations*, London, U.K., pp. 309-329, 1980.

[13] Kiriakidis K., Non-linear control system design via fuzzy modeling and LMIs, *International Journal of Control*, 72(7):676-685, 1999.

[14] Kothare M., Mettler B., Morari M., Bendotti P., Falinower C., Level Control in the Steam Generator of a Nuclear Power Plant, *IEEE Trans. On Control Systems Technology*, 8(1):55-69, 2000.

[15] Menon S.K. and Parlos A.G., *Gain-scheduled nonlinear control of U-tube steam generator water level*, Nuclear Sci. Eng., vol. 111, pp. 294-308, 1992.

[16] Morari M. and Lee J. H., Model predictive control: Past, present, and future, *Computers & Chemical Eng.*, pp. 667-682, 1999.

[17] Park G. Y., Seong P. H., Application of a self-organizing fuzzy logic controller to nuclear steam generator level control, *Nuclear Engineering and Design*, vol. 167, pp. 345-356, 1997.

[18] Pedrycz W. and Gomide, F., *Fuzzy Systems Engineering: Toward Human-Centric Computing*, Wiley-IEEE Press, 2007.

[19] Precup, R. E., Tomescu M., Preitl S., Lorenz System Stabilization Using Fuzzy Controllers, *International Journal of Computers Communications and Control*, 2(3):279-287, 2007.

[20] Ross T.J., *Fuzzy Logic with Engineering Applications*, second ed., Wiley & Sons, 2004.

[21] Vesselenyi, T; Dzitac, S; Dzitac, I, Manolescu M.J., Fuzzy and neural controllers for a pneumatic actuator, *International Journal Of Computers Communications & Control*, 2(4):375-387, 2007.

[22] Yager R., R., Zadeh L. A., *An Introduction to Fuzzy Logic Applications in Intelligent Systems*, Kluwer Academic Publishers, Norwell, MA, 1992.

[23] Yen J., Langari R., Zadeh L., A., *Industrial Applications of Fuzzy Logic and Intelligent Systems*, IEEE Press, Piscataway, NJ, 1995.

[24] Ying H., *Fuzzy Control and Modeling: Analytical Foundations and Applications*, Wiley-IEEE Press, 2000.

[25] Zadeh, L. , A New Frontier in Computation - Computation with Information Described in Natural Language, *International Journal of Computers Communications and Control*, 3(S):26-27, 2008.

[26] Zadeh L. A., Toward a generalized theory of uncertainty (GTU): an outline, *Information Sciences-Informatics and Computer Science: An International Journal*, 172(1-2):1-40, 2005.

[27] Zadeh, L. A., Knowledge Representation in Fuzzy Logic, *IEEE Transactions on Knowledge and Data Engineering*, 1(1):89-100, 1989.

[28] Zadeh, L. A., Is there a need for fuzzy logic?, *Information Sciences: an International Journal*, 178(13):2751-2779, 2008.

[29] Zadeh L.A., Tufis D., Filip F.G., Dzitac I.(Eds.), *From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence*, Editing House of Romanian Academy, Bucharest, ISBN 978-973-27-1678-6, 2008.

[30] Zhao F., Ou J., Du W., Simulation modeling of nuclear steam generator water level process a case study, *ISA Transactions*, 39, pp. 143-151, 2000.

Daniela Hossu is Associated Professor at University "Politehnica" of Bucharest. In 1996 she obtained M.S. degree in Electrical Engineering, Specialty - Control Systems and Informatics in Power Systems at University "Politehnica" of Bucharest. In 2002 she obtained a Ph.D. in Automatic Control from the University "Politehnica" of Bucharest. She was involved in several projects for retrofitting the Instrumentation & Control System of Thermal Power Plants. Since 1998 she has been involved in lecturing and teaching assistantships at University "Politehnica" of Bucharest in the following fields: automatic control, data transmission and data compression. Her research interests include Power Plant Control Systems, Artificial Intelligence, Industrial Communication, Data Compression, Software architectures for industrial control.

Ioana Făgărăşan received the M.S. (1996) degree in Electrical Engineering, Specialty - Control systems and Informatics in Power Systems, and the PhD (2002) degree in Automatic Control from the University Politehnica of Bucharest (UPB). Since 1996 she has been Assistant Professor, Lecturer and is currently Associated Professor in the department of Control Engineering and Industrial Informatics at the UPB. She has participated in research cooperation programs between the University and the Institute for Automatic Control, Technical University of Darmstadt, in the Fault Detection and Identification (FDI) field. Between 2002 and 2004 she worked as associate professor at the National Polytechnic Institute of Grenoble (Automatic Control Laboratory). Her research interests include control system for technical processes, fault detection and isolation methods, diagnostic algorithms and fault tolerant systems.

Andrei Hossu is Professor at University "Politehnica" of Bucharest. In 1982 he obtained M.S. in Electrical and Computer Engineering, Department of Automatic Control and Computers, The Polytechnic Institute of Bucharest, Romania. In 2002 he obtained a Ph.D. in Engineering Sciences, the Electrical control field from the Faculty of Automatic Control and Computers, the University "Politehnica" of Bucharest, Romania. His teaching interests include: Human-Machine Interface, Artificial Vision and Artificial Intelligence Methods. His research interests include: Multiprocessor systems for real time control, Industrial robots control systems, Algorithms, methods for artificial vision systems for industrial robots, Structured programming for industrial robots, Software architectures for industrial control.

Sergiu Stelian Iliescu has received the M.S. (1965) degree from the Power System Faculty, Automatic Control Department and Ph. D. (1979) degree in Computers- Electronics from University "Politehnica" of Bucharest (UPB). He was starting his professional activity in execution and implementation of automatic control equipments and systems, within the Automatic Control System Execution Company (IMIA). The next step was to work as design engineer within Power Systems Studies and Design Institute (ISPE). Since 1968 he is part of the teaching and research team from Automatic Control and Computers Faculty, UPB and acts as Full Professor since 1995. He has experience in process computers (TU Dresden, 1971-1972) and coordinates collaborative research activities with TU Darmstadt (1992-2004), in two research field: in system theory, modeling, identification and control technical processes (Institute of Automatic Technique, TUD) and in power system control and monitoring (Institute of Electrical Power Systems, TUD). His research and teaching domain includes: automatic control theory, informatics system for supervision and control of industrial processes with concern to power system field and system engineering analysis.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 2, pp. 193-204

Objects Detection by Singular Value Decomposition Technique in Hybrid Color Space: Application to Football Images

M.Moussa Jlassi, A. Douik, H. Messaoud

Mourad Moussa Jlassi
Ecole Nationale d'Ingénieurs de Monastir
Avenue Ibn El Jazzar - 5019 - Monastir - TUNISIE
E-mail: Mourad.enim@yahoo.fr

Ali Douik and Hassani Messaoud
Ecole Nationale d'Ingénieurs de Monastir
Avenue Ibn El Jazzar - 5019 - Monastir - TUNISIE
E-mail: Ali.douik@enim.rnu.tn
E-mail: Messaoud.hassani@enim.rnu.tn

Abstract: In this paper, we present an improvement non-parametric background modeling and foreground segmentation. This method is important; it gives the hand to check many states kept by each background pixel. In other words, generates the historic for each pixel, indeed on certain computer vision applications the background can be dynamic; several intensities were projected on the same pixel. This paper describe a novel approach which integrate both Singular Value Decomposition (SVD) of each image to increase the compactness density distribution and hybrid color space suitable to this case constituted by the three relevant chromatics levels deduced by histogram analysis. In fact the proposed technique presents the efficiency of SVD and color information to subtract background pixels corresponding to shadows pixels. This method has been applied on colour images issued from soccer video.
In the other hand to achieve some statistics information about players ongoing of the match (football, handball, volley ball, Rugby...) as well as to refine their strategy coach and leaders need to have a maximum of technical-tactics information. For this reason it is prominent to elaborate an algorithm detecting automatically interests color regions (players) and solve the confusion problem between background and foreground every moment from images sequence.

Keywords: Segmentation, Color Image, Statistic Algorithm, Histogram Analysis, Singular Value Decomposition.

1 Introduction

Image segmentation based on background modeling is a research thematic up until now locates on the head of enormous number of studies. Indeed background modeling and image segmentation techniques leads to remedy many problems in a wide spectrum of computer vision applications; K. Elgammal et al. [1] present non-parametric model for the images segmentation when the background is dynamic. Ying Ming et al. [2] propose a statistical algorithm inspired from the idea of Elgammal based on Cauchy distribution; they proved that the intensity values of background pixels are adapted to Cauchy's distribution. O. Javed et al. [3], propose mixture models to handle the backgrounds that exhibit multimodal characteristics, integration of gradient information are suggested as another feature of the multiple models. Although Gaussian mixture models can converge to any arbitrary distribution provided by enough number of components. R. Agarwal et al. [4] present a data-hiding algorithm that exploit the singular value decomposition (SVD) representation of the data. They compute the SVD of the host image and the watermark and embed the watermark in the singular vectors of the host image. The proposed method leads to an imperceptible scheme for digital images, both in grey scale and color and is quite robust against attacks like noise and JPEG compression.

In this paper, we describe an algorithm modeling and subtracting the background pixels based on SVD approach which is used in many computer vision problems on one hand, select the appropriate color space among the wide set of color levels commonly used in color image analyze on the other hand. Both compactness power of various density distributions and quality of approximated image are increased by singular value decomposition approach. Also this technique can be used to answer several purpose for example eliminate noise and highlights [5, 6] issues from the change of illuminations, dynamic background pixels, camera displacement and shadows. Finally specify the suitable color space constituted by the three significant color levels, where this segmentation technique will be realized and use a new similarity measure between reference and candidate image that consists to calculate the intersection coefficients of color histogram images.

This paper is organized as follows. Section 2 presents an overview of statistical algorithms and background model. In section 3 we define the singular value decomposition and we describe the main issues of the SVD approach. Hybrid color space constituted by significant levels has been explained in section 4. Finally, section 5 describes foreground segmentation and experimental results which will evaluate the robustness of this technique.

2 An overview of statistical algorithms and background model

2.1 Statistical Algorithms

A large variety of Background subtraction and image segmentation's algorithms have been developed in last few years ranging from parametric to non-parametric, from pixel to region [8]. Statistical algorithms are frequently used in computer vision; M. Seki et al. [7] propose background subtraction based on Cooccurrence of image variations. Oriol Pujol et al. [9] propose a new deformable model defined in a statistical framework to segment objects of natural scenes. They perform a supervised learning of local appearance of the textured objects and construct a feature space using a set of co-occurrence matrix measures. Linear Discriminant Analysis allows them to obtain an optimal reduced feature space where a mixture model is applied to construct a likelihood map. Instead of using a heuristic potential field, their active model is deformed on a regularized version of the likelihood map in order to segment objects characterized by the same texture pattern. Different tests on synthetic images, natural scene and medical images show the advantages of their statistic deformable model.

K. Verma et al. [10] propose a new improved mountain clustering technique, which is compared with some of the existing techniques such as K-Means, FCM, EM and Modified Mountain Clustering. The performance of all these clustering techniques towards color image segmentation is compared in terms of cluster entropy as a measure of information and observed via computational complexity. The cluster entropy is heuristically determined, but is found to be effective in forming correct clusters as verified by visual assessment. A. Farhadi et al. [11] present a method for the segmentation of images based on local higher order statistics. The algorithm can be applied for the separation of objects from a texture background and the segmentation of textures. The proposed technique makes no use of a data bank and its complexity is $O(\chi)$ where χ is the number of pixels.

2.2 Non-parametric Background Model

Because the parametric background model still lacks flexibility when the background pixels are dynamic, a highly flexible non-parametric technique is proposed to estimate background probabilities from many recent samples over time using Kernel density estimation. In the non-parametric model all recently observed pixel values $x_1, x_2,.. x_N$ are modeled by probability density functions using a certain kernel estimator function, which is often chosen to be a Gaussian. The weighted sum of all these Gaussians results in the final probability density function of the pixel value x_t :

$$P(x_t) = \frac{1}{n}\sum_{i=1}^{N} K(x_t - x_i) \tag{1}$$

The kernel estimator function K is chosen as a Normal function $N(0, \Sigma)$ where Σ is the kernel function bandwidth. Color channels are assumed independent and each channel has its own kernel band width σ^2. These assumptions lead us to the final density estimation which can be written as:

$$Pr(x_t) = \frac{1}{K}\sum_{i=1}^{k}\prod_{j=1}^{d} \frac{1}{\sqrt{2\pi\sigma_j^2}} \exp - \frac{(x_{tj} - x_{ij})^2}{2\sigma_j^2} \tag{2}$$

When this probability is higher than a certain threshold which is a global threshold over all the image, the pixel is classified as a foreground pixel. In other applications using kernel density estimation the kernel bandwidth dependent on the number of samples. To estimate the kernel band width σ^2 for the j^{th} color channel for a given pixel we compute the median absolute deviation over the sample for consecutive intensity values of the pixel. That is, the median, m, of $|x_i - x_{i+1}|$ for each consecutive pair (x_i, x_{i+1}) in the sample, is calculated independently for each color channel. If we assume that the distribution is Normal $N(\mu, \sigma^2)$, then the deviation $(x_i - x_{i+1})$ is Normal $N(0, 2\sigma^2)$. So the standard deviation of the first distribution can be estimated by

$$\sigma = \frac{m}{0.68\sqrt{2}} \tag{3}$$

This method ensures that the local deviation is large when there are many large jumps between consecutive samples and smaller when this is not the case.

3 Singular Value Decomposition (SVD)

3.1 Introduction

The singular value decomposition (SVD) is a generalization of the Eigen decomposition which can be used to analyze rectangular matrices (the Eigen-decomposition is defined only for squared matrix). By analogy with the Eigen decomposition, which decomposes an image into two simple matrix, the main idea of the SVD is to decompose a rectangular matrix as a product three matrix: Two orthogonal and one diagonal. The SVD is equivalent to principal component analysis (PCA) and is therefore an essential tool for multivariate analysis.

3.2 SVD Image Approximation

The SVD uses Eigen decomposition of a positive semi-definite matrix in order to derive a similar decomposition applicable to all rectangular real number matrix. The main idea is to decompose any matrix into three matrix; two orthonormales and one diagonal. Formally, if **A** is a rectangular matrix, its SVD is written by:

$$A = P\Delta Q^T \tag{4}$$

*P: the (normalized) eigenvectors of the matrix AA^T ($PP^T = I$). The columns of **P** are called the left singular vectors of **A**.*
*Q: the (normalized) eigenvectors of the matrix AA^T ($QQ^T = I$). The columns of **Q** are called the right singular vectors of **A**.*
Δ the diagonal matrix of the singular values, $\Delta = \Lambda^{\frac{1}{2}}$ with Λ being the diagonal matrix of the Eigen values of matrix AA^T and of the matrix $A^T A$, since they are the same.

A non-parametric background modeling technique has been applied on soccer video images. The main problem that can appear is the occurring of wrong detection pixels. Indeed the detection of moving objects as a shadows pixels allows to an over segmentation which will damage many works where this paper is registered. This algorithm is extremely important because it is a part of player's classification and tracking [15] on a football scenes.

SVD representation is very useful in image processing applications in particular for the spectral image compression [16]. The interest of SVD in this method is to determine number's SVD of the treated image. Besides we will see the prominent contribution using SVD approach to restore and eliminate shadows, highlights and noise from camera displacement.

In the developed background segmentation method described in this paper, the main objective is summarized by the use of the singular value decomposition. Let A is a given image represented by a matrix $A_p = [a_{ij}]$, which can be decomposed into a product of three matrix $U_k S_k V_k^T$, where a_{ij} is the appearance frequency of background pixel's chromaticity and intensity; (p= red, green, blue).

D. Valentin et al. [17] and H. Abdi et al. [18] prove the property's importance to approximate a given matrix by SVD which will be substituted by another having reduced rank and dimension of each initial level from r to k (i.e. suppression of r-k column). The determined singular values for each level was presented in frequency space, their representation prove that for each one correspond a discrete frequency. The noise that can appear in the signal (in frequency space the amplitude of noise is constant) corresponds to a low amplitude, singular value whereas high amplitudes of these represents global signal energy.

3.3 Confidence intervals research

This section presents the steps which allow to determine the singular values number for each level.

- Image improvement : In this section, we describe the basic background model and the background subtraction process with singular value decomposition. The latter was used in both restoration, reconstruction of considred image, and increase the compactness distribution of different class, and also provide useful image information.

- Weights Interpretation: To evaluate mathematical contribution of singular value decomposition, a quantification of global signal energy distribution according the weight of each singular value S_{kk} was done. Figure 1 illustrate the energy distribution E defined by:

$$E = \sum_{i=1}^{k} A_i^2 \tag{5}$$

The relative energy contained by each singular value K, noted p_k is defined by:

$$P_k = \frac{S_{kk}^2}{E} \tag{6}$$

Where the energy of the K singular value is equal to S_{kk}^2.

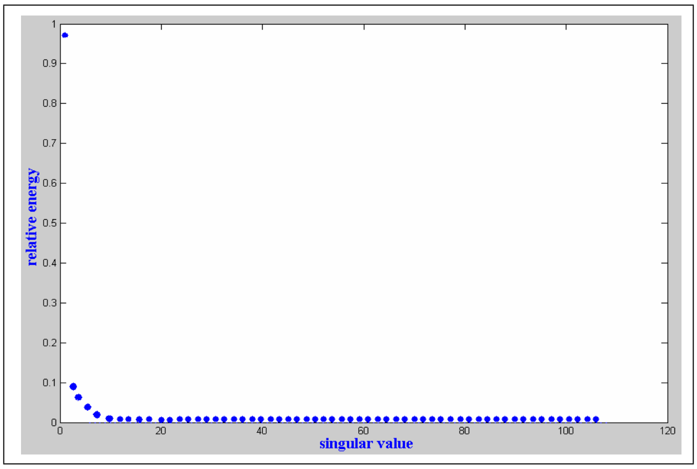


Figure 1: Evaluation of energy according to singular values decomposition.

As it is shown on the figures 2a, 2b and 2c, the choice of the image's size which will be manipulated, is deduced from the curves representing standard deviation of each colour levels according to the singular value decomposition. In fact a good choice of singular value number leads to reduce both compactness in different distribution and computing time.

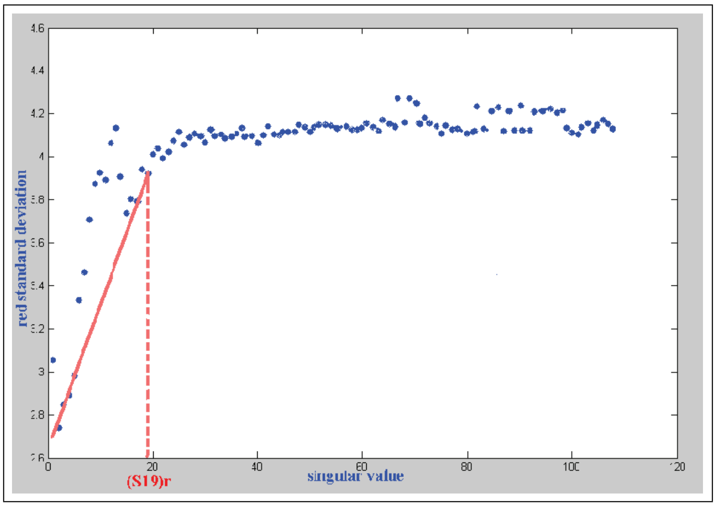


Figure 2: Evaluation of standard deviation according to singular values of the red level.

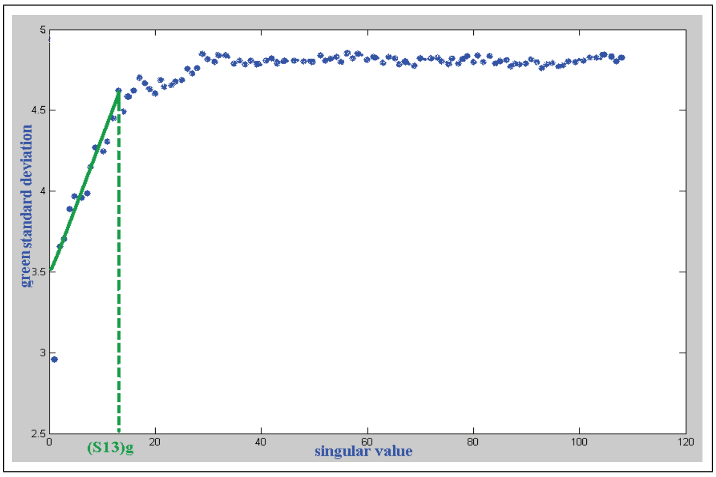


Figure 3: Evaluation of standard deviation according to singular values of the green level.

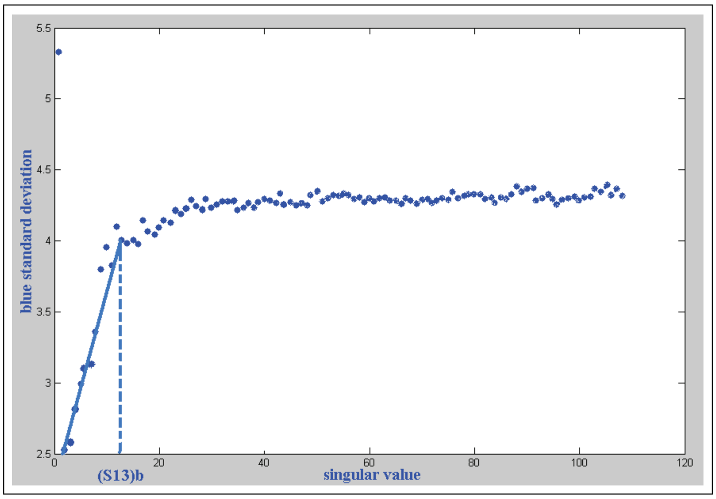


Figure 4: Evaluation of standard deviation according to singular values of blue level.

Figures 2a, 2b and 2c denote two zones, the first one defined in the interval [0, $(S_{kkl})_i$ where $(S_{kkl})_i$ is the singular value limits corresponding to the linear part of the curve i(red, blue, green), on this zone the curve prove a slope, beyond $(S_{kkl})_i$ a second zone appears where the standard deviation varied slightly, therefore the optimal singular value $(\hat{S}_{kkl})_i$ must necessarily belong to the first zone of each curve. Table 1 illustrates initial and improved standard deviations for three channels (RGB).

- Choice of the singular values number: the choice of the singular values number, which will be kept, depends on two issues, the first one is the energy curve evaluated by figure 2 and the second is standard deviation curves of each chromatic level shown in figures 2a, 2b and 2c. In fact we specify for each component the limit singular value defined previously. Table 2 shows confidence intervals as well as singular value limits and the optimal value.

Table 1: Evaluation of improvement parameters.

	σ	μ
R	4.2044	119.610
G	4.7227	152.080
B	4.313	88.988
R_{svd}	3.927	119.07
G_{svd}	4.620	152.05
B_{svd}	3 995	88.407

Table 2: Specification of confidence intervals.

	$(S_{kkl})_i$	$(\hat{S}_{kk})_i$	confidence intervals
R	29	19	[0, 29]
G	28	13	[0, 28]
B	19	13	[0, 13]

4 Hybrid color space research by histogram analysis

Many descriptors can be used in image processing field, C. Huang et al. [12] combine texture and color descriptors [19] to form hybrid visual feature index to retrieve natural color images, in fact this method is insensitive to image rotation and translation and then the experimental results show that the method achieves better performance than other recent relevant methods. The RGB space isn't always the best one [20], Indeed, other colorimetric components, deduced from Red, Green and Blue, can be more suitable according to a considered case. Michael K. Ng et al. [13] consider restoring a single-color image from two degraded frames of the same scene by a RGB sensor and a luminance sensor. The RGB-to-YIQ transformation, the classical Tikhonov regularization and the Neumann boundary condition are used in the restoration process. R. Missaoui et al. [14] illustrate the superiority of an efficient content-based image mining and retrieval approach towards similarity analysis and retrieval effectiveness computation both in the use L* C* H* and CIECAM02 color spaces.

In order to refine the result, a new effectiveness measure are proposed and experimented on an image, this approach consists to convert image from RGB space to other one which are grouped in Ψ set and select the suitable system which minimizes the overlapping between different clusters. where $\Psi = (R_{RGB}$, G_{RGB}, B_{RGB}, r_{rgb}, g_{rgb}, b_{rgb}, H_{HSL}, S_{HSL}, L_{HSL}, X_{XYZ}, Y_{XYZ}, Z_{XYZ}, L_{Lab}, a_{Lab}, b_{Lab}, Y_{YUV}, U_{YUV}, V_{YUV}, C_{CMY}, M_{CMY}, Y_{CMY}, $I1_{I_1I_2I_3}$, $I2_{I_1I_2I_3}$, $I3_{I_1I_2I_3}$, $A_{AC_1C_2}$, $C1_{AC_1C_2}$, $C2_{AC_1C_2}$, Y_{YIQ}, I_{YIQ}, Q_{YIQ}, l_{lch}, c_{lch}, h_{lch}, H_{HSV}, S_{HSV}, V_{HSV}, Y_{YCbCr}, Cb_{YCbCr}, Cr_{YCbCr}, Y_{YPbPr}, Pb_{YPbPr}, Pr_{YPbPr}, Y_{YDbDr}, Db_{YDbDr}, $Dr_{YDbDr})$, multidimensional space defined by a set of color levels frequently used. A new similarity measures which compare two color histograms and calculates the intersection coefficients, as illustrates in figure 3a and 3b.

However Histogram analysis and conversion of image in various color space are an important technique to achieve many studies in this field and lead to make decision about discrimination capacity of these levels thus determine a hybrid color space constituted by the three significant components. Algorithm 1 calculate similarity criterion between reference, and candidate image.

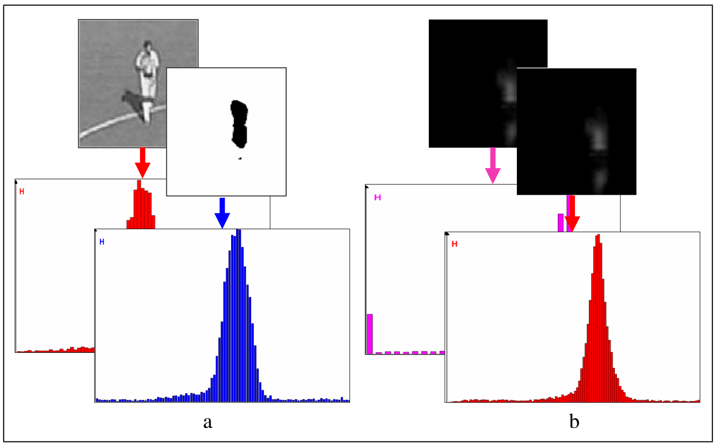


Figure 5: (a) represents the intersection coefficient (0.041965) between reference level and b_{Lab} for player A. (b) represents the intersection coefficient (0.019788) between reference level and H_{HSL} for player B.

Algorithm 1:

```
Read original image in gray level.
Calculate Histogram $H_g$.
  For each level i
    Read the image in this level
      Calculate Histogram $(H_p)_i$
        Estimate intersection between $H_g$ and $(H_p)_i$
          If inter $\approx$ 0
              $p_i$ is a suitable plan
            Else
              Choose $p_{i+1}$
          End if
  End For.
```

This algorithm is compiled for each player with various levels, the intersection coefficients between original (gray level) and converted frame were evaluated and given the following results:

- Relevant levels for player A: H_{HSL}, v_{Luv}, I_3 and I_2, b_{Lab}, r_{rgb} and b_{rgb}.
- Relevant levels for player B: H_{HSL}, I_3, Pr_{YPbPr}, $JPEGCr_{JPEGYCbCr}$, Dr_{YDbDr}, V_{YUV} and $C2_{YC1C2}$.

To appreciate and highlight the interest of this technique, we recapitulate results of some favorite levels in tables 3 and 4. This supervised training enables to control statistical parameters, confirm the efficient choice of this method and increase compactness power. Table 5 and 6 show the improvements introduced to the distribution of each different class.

- For player A: Table 5 illustrates mean μ and standard deviation σ before and after segmentation, of original image (gray level) and represented image in each relevant level and the ratio of the standard deviation σ, before and after segmentation, by standard deviation of reference image. σ_{refbef} (before) =15.232, σ_{refaft} (after) = 8.317.

Table 3: Significant levels for player A (P_A).

	level	inter
1	H_{HSL}	0
2	v_{Luv}	0
3	I_3	0.00022401
4	I_2	0.00029869
5	b_{Lab}	0.00171740
6	r_{rgb}	0.03345300
7	b_{rgb}	0.04196500

– For player B: Similarly for player B and with the same way, segmentation parameters were given in table 6 and we conserve the same notation given for player A.

Table 4: Significant levels for player B (P_B).

	level	inter
1	$JPEGCr_{JPEGYCbCr}$	0
2	I_3	0
3	H_{HSL}	0.00067204
4	Dr_{YDbDr}	0.00238950
5	Pr_{YPbPr}	0.00418160
6	V_{YUV}	0.01493400
7	$C2_{YC_1C_2}$	0.01978800

This technique contributes to research a suitable hybrid color space that is separate some distributions corresponding for each set pixels class by an iterative selection procedure. This procedure consists to extract the significant level from a set of color components according specified actors present in image. The obtained results show clearly how the use of color improves the segmentation quality.

5 Experimental results

Using the probability $Pr(x_t)$ calculated in the equation (3), a pixel is considered as a foreground pixel if $Pr(x_t) < th$. The threshold th is a global threshold over all the image that can be adjusted to achieve a desired percentage of false positives. Practically, this probability can be calculated in a very fast way using pre calculated lookup tables for the kernel function values given the intensity value difference,$(P_i - p_t)$, and the kernel function bandwidth. The detection of shadows as foreground pixels was a source of confusion with a background. The solution that allows to overcome this problem incorporates both singular value decomposition to increase the compactness power of distributions and color information [21] where pixels intensities will be expressed. Indeed chrominance levels were better than these of lightness. They lead to discriminate foreground and their shadows. The work space where this segmentation was carried out is the **Hrb** constituted by **HSL** and **rgb** Previously determined.

Table 5: Evaluation of parameters in favorite levels before and after segmentation (P_A).

	σ_{bef}	Rap_{bef}	σ_{aft}	Rap_{aft}
H_{HSL}	14.353	0.942	2.804	0.099
v_{Luv}	3.873	0.254	2.480	0.087
I_3	5.477	0.359	4.370	0.154
I_2	5.831	0.382	7.424	0.262
b_{Lab}	6.082	0.399	5.624	0.198
r_{rgb}	10.392	0.682	20.640	0.728
b_{rgb}	11.705	0.768	13.935	0.492

Table 6: Evaluation of parameters in favorite levels before and after segmentation (P_B).

	σ_{bef}	Rap_{bef}	σ_{aft}	Rap_{aft}
$JPEGCr$	16.000	1.076	0.747	0.027
I_3	5.385	0.362	4.502	0.164
H_{HSL}	16.000	1.076	24.416	0.894
Dr_{YDbDr}	6.000	0.403	0.811	0.029
Pr_{YPbPr}	9.0554	0.609	0.765	0.028
V_{YUV}	10.050	0.676	0.762	0.027
$C2_{YC1C2}$	10.724	0.721	0.760	0.027

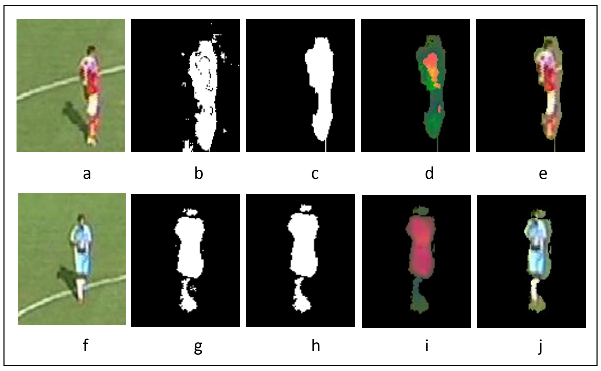


Figure 6: (a, f) original images. (b, g), (c, h) present respectivly binary images before and after treatment. (d, i) segmented image with Hrb system. (e, j) output images in RGB system.

6 Conclusion

In this paper, we present a non-parametric kernel density estimation technique, to model and subtract background which was developed in other color space more suitable instead the RGB space. Experimental results has been improved by using an appropriate mathematical approach for objects detection and shadows suppression.

This technique incorporates singular value decomposition to increase the compactness power of each distribution, also approximate a new image which will be manipulated on one hand, determine a suitable color space constituted by significant levels among set of levels commonly used in color image analysis based on a novel similarity measures for comparing color histogram of reference and candidate image where we calculate the intersection coefficient which present criterion's discrimination on other hand. Also this technique proves that the gray levels and the RGB space not efficient for all applications. However the deduced color space (Hrb) allows to convincing results and confirm the efficiency of this method.

Bibliography

[1] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, Background and Foreground Modeling Using Nonparametric Kernel Density Estimation for Visual Surveillance, *Proc. IEEE* 90(7), pp. 1151-1163, 2002.

[2] Y. Ming, J. Jiang, J. Ming, Background Modeling and Subtraction Using a Local-Linear-Dependence-Based Cauchy Statistical Model, *Proc. VIIth Digital Image Computing: Techniques and Applications*, pp. 469-478, 2003.

[3] O. Javed, K. Shafique, and M. Shah, A hierarchical approach to robust background subtraction using color and gradient information, *In IEEE Workshop on Motion and Video Computing*, 2002.

[4] R. Agarwal and M. S. Santhanam, Digital watermarking in the singular vector domain, *International Journal of Image and Graphics*, Vol. 8, No. 3, pp. 351-368, 2008.

[5] YL. Liu, J. Wang, X. Chen, A robust and fast non-local means algorithm for image denoising, *Journal of Computer Science and Technology*, 23(2): 270-279 Mar. 2008.

[6] X. Zhang, K. Kobayashi, S. Saito and M. Nakajima, Reflectance-Field-Based Separation of Surface Reflection Components, *Information and Media Technologies*, Vol. 1, No. 2, pp.1040-1048, 2006.

[7] M. Seki, T.F. Wada and H.Sumi, Background Subtraction Based on Cooccurrence of Image Variations, *In Computer Vision and Pattern Recognition*, pp. 65-72, 2003.

[8] T. Brox, M. Rousson, R. Deriche, and J.Weickert, Unsupervised segmentation incorporating colour texture and motion,*Computer Analysis of Images and Patterns*, volume 2756 of Lecture Notes in Computer Science, pp. 353-360, 2003.

[9] O. Pujol and P. Radeva, Segmentation by Statistical Deformable Models, *International Journal of Image and Graphics*, Vol. 4, No. 3, pp. 433-452, 2004.

[10] K. Verma and M. Hanmandlu, Color Segmentation via Improved Mountain Clustering Technique, *International Journal of Image and Graphics*, Vol. 7, No. 2, pp. 407-426, 2007.

[11] A. Farhadi, M. Shahshahani, Image segmentation via local higher order statistics, *International Journal of Imaging Systems and Technology*, Vol. 13, No. 4, pp. 215-223, 2003.

[12] C. Huang, J. Zhou and S. Yu, Color Image Retrieval Based on Color-Texture-Edge Feature Histograms, *International Journal of Image and Graphics*, Vol. 6, No. 4, pp. 583-598, 2006.

[13] Michael K. Ng and N. K. Bose, Fast color image restoration with multisensors, *International journal of imaging systems and technology*, Vol. 12, No. 5, pp. 189-197, 2002.

[14] R. Missaoui, M. Sarifuddin and J. Vaillancourt, Similarity measures for efficient content-based image retrieval, *IEE Proceedings. Vision, Image, and Signal Processing*, Vol. 152, No. 6, pp. 875-887, 2005.

[15] A. Colombari, A. Fusiello, V. Murino, Segmentation and tracking of multiple video objects, *Pattern Recognition*, pp. 1307-1317, 2007.

[16] H. KOTERA, RGB to Pseudo-Spectral Image Conversion Using Spectral Palette and Compression by Singular Value Decomposition, *NIHON GAZO GAKKAISHI*, Vol. 42, pp. 215-223, 2003.

[17] Abdi, H. (2003). Multivariate analysis. In M. Lewis-Beck, A. Bryman, and T. Futing (Eds): "Encyclopedia for research methods for the social sciences". Thous and Oaks: Sage.

[18] Abdi, H., Valentin, D. (2006). "Mathématiques pour les sciences cognitives (Mathematics for cognitive sciences)". Grenoble: PUG.

[19] H. Matsuda, S. Kubota and H. Sato, Comparison of Hair Color Using Image Analysis, *Japanese Journal of Forensic Science and Technology*, Vol. 13, No. 2, pp. 151-166, 2008.

[20] M. Rousson, T. Brox, and R. Deriche, Active unsupervised texture segmentation on a diffusion based feature space, *In Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 699-704, June 2003.

[21] M. G. Linguraru, M. Á. G. Ballester, N. Ayache. Deformable Atlases for the Segmentation of Internal Brain Nuclei in Magnetic Resonance Imaging, *International Journal of Computers, Communications and Control*, Vol. II, No. 1, pp. 26-36, 2007.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 2, pp. 205-215

On the Use of the FuzzyARTMAP Neural Network for Pattern Recognition in Statistical Process Control using a Factorial Design

J. A. Vazquez-Lopez, I. Lopez-Juarez, M. Peña-Cabrera

J. A. Vazquez-Lopez
Instituto Tecnologico de Celaya
CIATEC, A.C. Posgrado Interinstitucional en Ciencia y Tecnologia (PICyT)
Industrial Engineering Department
Celaya, Gto. MEXICO
E-mail: antoniov@itc.mx

I. Lopez-Juarez
Centro de Investigacion y de Estudios Avanzados del IPN - Unidad Saltillo
Robotics and Advanced Manufacturing Department
Carretera Saltillo-Monterrey Km 13. Ramos Arizpe, Coahuila. MEXICO
E-mail: ismael.lopez@cinvestav.edu.mx

M. Peña-Cabrera
National Autonomous University of Mexico
Engineering Computing Science and Automation Department
Institute of Research on Applied Mathematics and Systems
Mexico, City. MEXICO
E-mail: mario@leibniz.iimas.unam.mx

Abstract: Time-series statistical pattern recognition is of prime importance in statistics, especially in quality control techniques for manufacturing processes. A frequent problem in this application is the complexity when trying to determine the behaviour (pattern) from sample data. There have been identified standard patterns which are commonly present when using the X chart; its detection depends on human judgement supported by norms and graphical criteria. In the last few years, it has been demonstrated that Artificial Neural Networks (ANN's) are useful to predict the type of time-series pattern instead of the use of rules. However, the ANN control parameters have to be fixed to values that maximize its performance. This research proposes an experimental design methodology to determine the most appropriate values for the control parameters of the FuzzyARTMAP ANN such as: learning rate (β) and network vigilance (ρ_a, ρ_b, ρ_{ab}) in order to increment the neural network efficiency during unnatural pattern recognition.

Keywords: Statistical Process Control, Control Charts, Artificial Neural Network (ANN), FuzzyARTMAP, and Factorial Design.

1 Introduction

To preserve product quality an accurate knowledge of the production process is necessary. This requires the automation of quality control systems and the use of control charts as introduced by Dr. Walter A Shewart to observe the behaviour of the manufacturing process.

Control charting is the key point in Statistical Process Control (SPC) implementation. The correct application of these Control Charts requires satisfying statistical assumptions such as the independence of the random variable and symmetry in its probability distribution [1]. If these assumptions are met then the use of Control Charts is correctly applied since the Upper and Lower limits are established as $\pm 3\sigma$ from the global mean of the X random variable. In figure 1, the probability distribution of X is shown under both circumstances with symmetry and without symmetry.

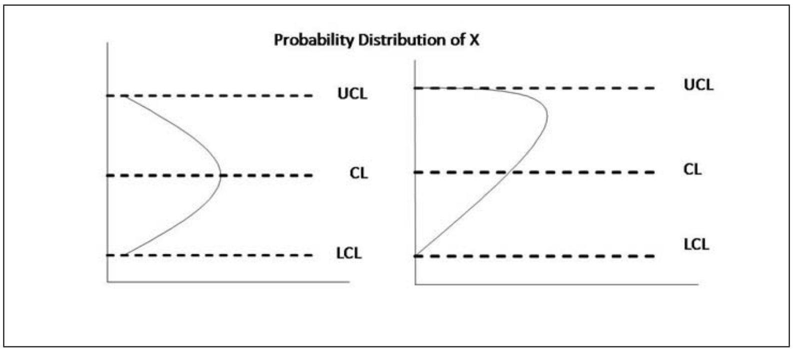


Figure 1: X Symmetry importance to establish the control limits

The power of an X chart is based on its capacity to differentiate special and natural causes of variation; however, important disadvantages on the use of this form of quality control exist because traditional control chart with control limit can only indicate when to seek a disturbance and not where and what to look for, generating then hurried and sometimes mistaken diagnostics [2].

Trying to know where and what has happened in the manufacture process may be possible by using pre-established rules in combination with human judgment. These rules are commonly referred to as: points outside the control limits, run of consecutive points, non-random patterns and points near the control limits [1]. The efficiency of the use of these rules has been investigated and it has been found that is not enough to recognise the type of statistical pattern ([3], [4], [5], [6], [7], [8], [9], [10], [11], [12]), which would give the correct answer to the questions of where and what to look for. This is why researchers suggest the use of the neural networks as an alternative approach to identify variation data in statistical patterns [4].

A neural network is a soft computing system [25], which consists of a number of elements (nodes) strongly interconnected that have the ability to process information as a result of a process of dynamic work of those nodes and connections to external points to the network [13]. Neural networks are efficient in recognizing data variation [2], [5], [14], especially in asymmetric probability distributions [2]. Among the existing neural networks, the Fuzzy ARTMAP Network is widely recognized due to its on-line and fast learning capability for pattern recognition tasks [15], [16].

For many processes of manufacture, the parameters can be obtained with neural network, with exception in many complex biotechnological processes. In these cases, [23] proposed the use of grey-box models which combine a priori knowledge expressed in terms of a white-box model, with a black-box model such as neural network, and [24] developed a Matlab®Toolbox for the construction of grey-box neural network models.

This paper is organised as follows: In the next section an introduction to the ART theory is given first followed by the standardisation and coding algorithm for the input data in the neural network using the Monte Carlo method. Section 3 describes the statistical pattern generation. Training and testing results are provided in Section 4. Section 5 describes the factorial design for the selection of the network parameters as well as the experimental results. An analysis for the pattern variation findings is given in Section 6 while Conclusions and Future work are given in Section 7.

2 Adaptive resonance theory

The Adaptive Resonance Theory (ART) [17] was developed by Stephen Grossberg and Gail Carpenter at Boston University to solve the called stability-plasticity dilemma. That is, the system is sensitive to novelty capable of distinguishing between familiar and unfamiliar events (plastic) and still remains stable. Different model variations have been developed to date based on the original ART-1 algorithm for binary input patterns [18], ART 2-A for analogue and binary input patterns [19], and ART 3 based on chemical transmitters. Supervised learning is possible through ARTMAP [20] that uses two ART modules and its variants, Fuzzy ARTMAP [16], Gaussian ARTMAP [21] and ART-EMAP even though there are many other variants adapted for specific applications [22]. In the next section a brief explanation of the mechanics of ART-1 and Fuzzy ARTMAP is given.

2.1 ART-1

The ART-1 architecture consists of two parts: attentional subsystem and orienting subsystem as illustrated in figure 2. The attentional subsystem is made up of two layers of nodes F_1 and F_2. In an ART network, information in the form of processing-element output reverberates back and forth between layers. If a stable resonance takes place learning or adaptation can occur. On the other hand, the orienting subsystem is in charge of resetting the attentional subsystem when an unfamiliar event occurs.

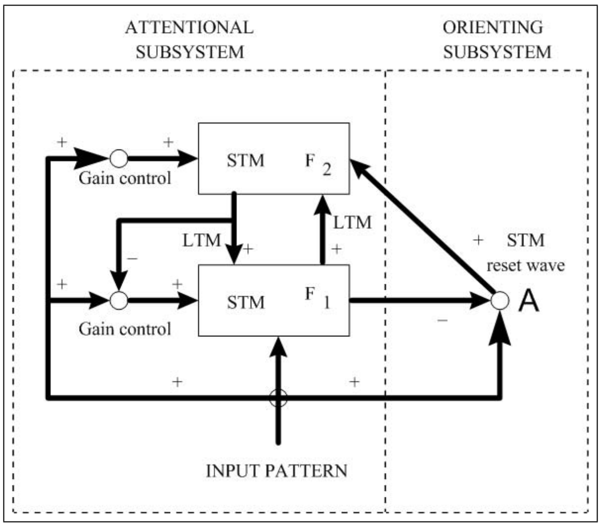


Figure 2: Basic ART Architecture

A resonant state can be attained in one of two ways. If the network has learned previously to recognise an input vector, then a resonant state will be achieved quickly when that input vector is presented. During resonance, the adaptation process will reinforce the memory of the stored pattern. If the input vector is not immediately recognised, the network will rapidly search through its stored patterns looking for a match. If no match is found, the network will enter a resonant state whereupon the new pattern will be stored for the first time. Thus, the network responds quickly to previously learned data, yet remains able to learn when novel data is presented, hence solving the so-called stability-plasticity dilemma. The activity of a node in the F_1 or F_2 layer is called short-term memory (STM) whereas the adaptive weights are called long-term memory (LTM). Gain controls handle the discrete presentation of the input signals. A vigilance parameter measures how much mismatch is tolerated between the input data and the stored patterns, which can be used to control the category coarseness control of the classifier.

2.2 Fuzzy ARTMAP

In the Fuzzy ARTMAP (FAM) network there are two modules ART_a and ART_b and an inter-ART module $Map-field$ that controls the learning of an associative map from ART_a recognition categories to ART_b categories. This is illustrated in figure 3.

The Map field module also controls the match tracking of ART_a vigilance parameter. A mismatch between Map field and ART_a category activated by input I_a and ART_b category activated by input I_b

increases ART_a vigilance by the minimum amount needed for the system to search for, and if necessary, learn a new ART_a category whose prediction matches the ART_b category. The search initiated by the inter-ART reset can shift attention to a novel cluster of features that can be incorporated through learning into a new ART_a recognition category, which can then be linked to a new ART prediction via associative learning at the $Map-field$.

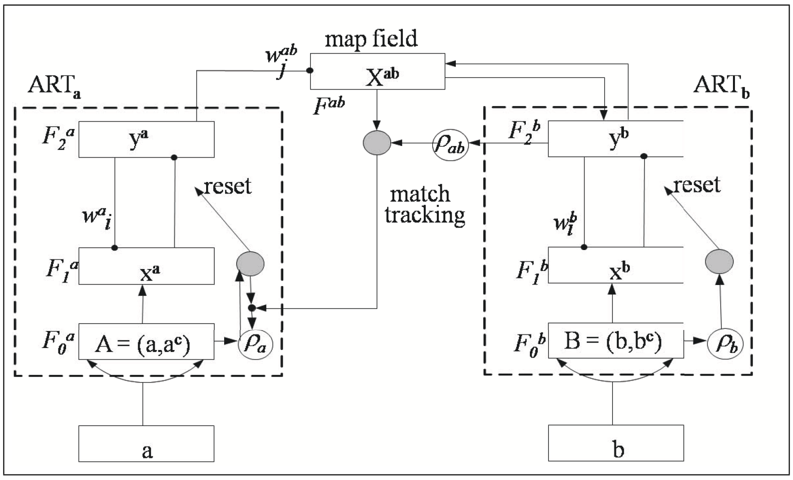


Figure 3: FuzzyARTMAP Architecture

A vigilance parameter measures the difference allowed between the input data and the stored pattern. Therefore this parameter is determinant to affect the selectivity or granularity of the network prediction. For learning, the FuzzyARTMAP has 4 important factors: Vigilance in the input module (ρ_a), vigilance in the output module (ρ_b), vigilance in the Map field (ρ_{ab}) and learning rate (β). These were the considered factors in this research.

2.3 Standardization and codification

The use of the neural network requires two important mathematical considerations, which are the standardization and the codification of the input data [2]. The training and testing data needs to be pre-processed in these two stages. The standardization means that the data have to be linearly transformed from data with mean (μ) and standard deviation (σ) into data with μ = 0 and σ = 1 using equation 1, as a result, the sample data is within the interval (-3.9, +3.9).

$$Y_t = \left(\frac{x_t - \mu}{\sigma}\right) \tag{1}$$

where:
x_t = sample value at sampling time t.
Y_t = standardized value from x_t.
μ = process mean.
σ = process standard deviation.

The x_t data are generated by a process simulator of Monte Carlo, according to equation 2

$$x_t = \mu + n_t + d_t \tag{2}$$

where:
μ = process mean.
n_t = common cause variation at sampling time t.

d_t = special disturbance at time t ($d_t = 0$ when the pattern is natural).

Shift.

$$d_t = ud \tag{3}$$

where:
u = parameter to determine the position of shifting (0: before shifting; 1: after shifting).
d = displacement of mean in terms of σ.

Trend slope.

$$d_t = st \tag{4}$$

where:
s = trend slope in terms of σ.
t = sampling time.

On the other hand, with the codification of Y_t the variation interval of [0,1] is obtained, which is a requirement for the neural network operation that reduces the effects of common causes of variation (noise) [2]. The codification of x_t considered the interval [-7.625, 7.625], whose range is greater to the expected Y_t range.

3 Pattern data generation

A specific value x_t of sample data is obtained from the sum of three mathematical considerations:

- Global and historical effect (μ).
- Natural variation effect (n_t).
- Disturbance variation effect (d_t).

Mathematically, equation 2 expresses this situation. In terms of industrial quality, these effects can be thought of as the global and historical mean obtained from experience (i), thought of as data variation which is unavoidable and it is always present (ii); and finally, the data variation due to disturbances which is associated to special causes that may cause the process to be out of statistical control (iii).

When a sample data has only influence on natural causes of variation, then $n_t > 0$ and $d_t = 0$, and the pattern data will be natural. On the other hand, if $d_t > 0$, then the pattern data will be unnatural, and it means that a cause of special variation has occurred in time t. It must be noticed that $0 < n_t < d_t$ for any type of special pattern data. If the d_t value is very similar to n_t then neural network output can be misleading between a special pattern and a natural one.

3.1 Natural Pattern

The data used for this pattern were generated using the Monte Carlo simulator using equation 2 with μ = 0 and $\sigma = 1$. An example of this type of pattern is shown in figure 4. The graph data comes from a time-series of X that did not consider any trend or shift in the global mean and with data distribution randomly assigned.

3.2 Shift Pattern

Data used for either downward shift or upward shift shows two data set separated by an abrupt change as shown in figure 5. This occurs because the reference mean also changes. This can be positive or negative and its magnitude depends on the special variation cause in the manufacturing process.

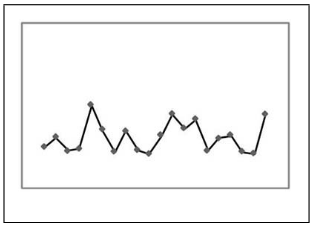


Figure 4: Natural Pattern

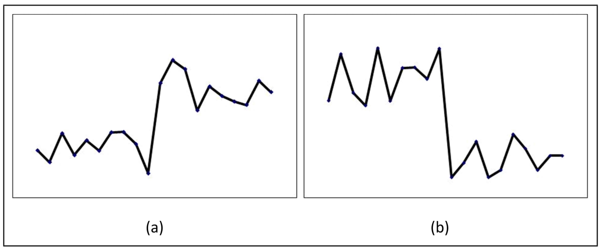


Figure 5: Shift Patterns (a)Upwards (b)Downwards

3.3 Trend Pattern

The type of pattern can be distinguished at a glance due to its upward or downward trend. In terms of mathematically what happens is that for any X_t in the data series (where t is not the last data), there will be points in time $t+1, t+2, t+3, \ldots, t+n$ of higher magnitude (upward trend) or lower magnitude (downward trend). Through linear regression is always possible to find out the slope magnitude which is also the magnitude of the effect that caused the special variation. This type of pattern can be observed in figure 6.

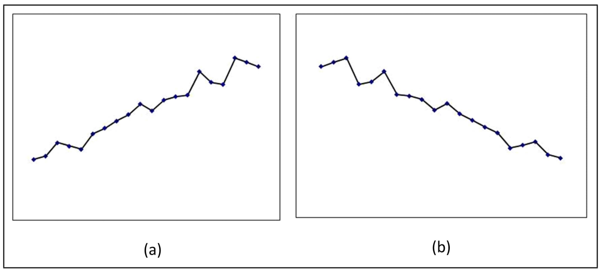


Figure 6: Trend Patterns (a)Upwards (b)Downwards

4 Training and testing

Five pattern types with diverse effects of special variation as mentioned above were studied. Table 1 shows the corresponding information indicating the used values for d_t and the output binary code used during the training and testing phase. The number of patterns for the input vector was 51 for training and 1,350 for testing.

5 Factorial design

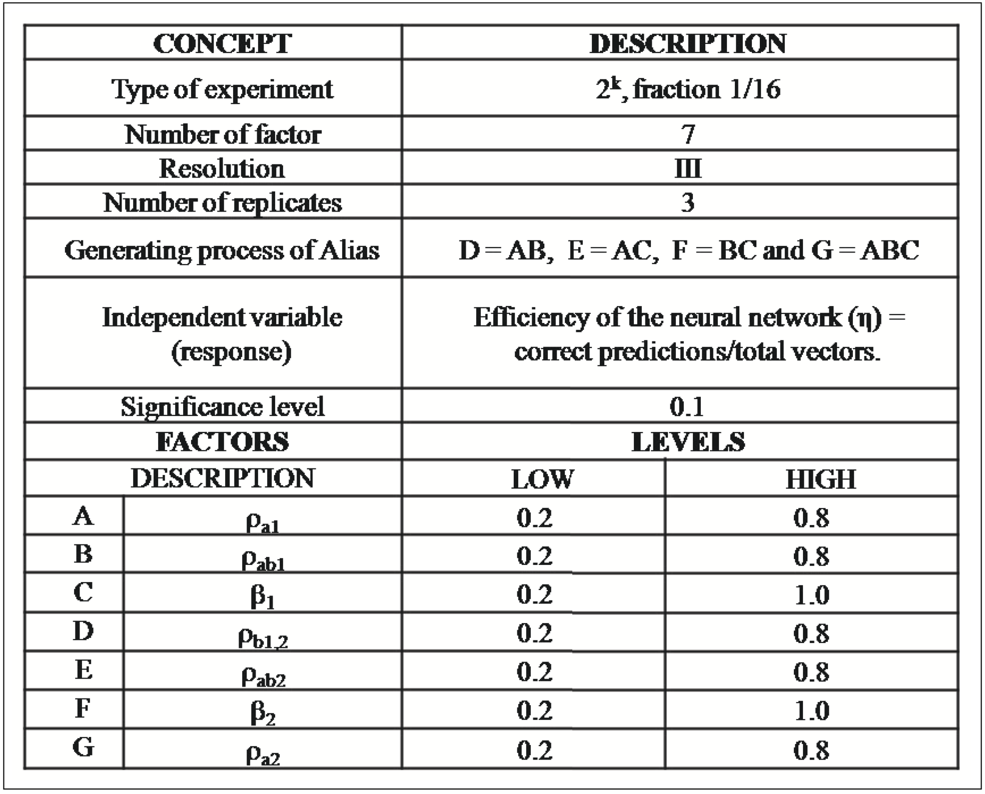
The experimentation required 120 tests based on 8 runs and 3 replicates. The information for the experimental design is shown in table 2. An analysis of variance from multiple experiments revealed the

Table 1: d_t values considered for each pattern type and output vector code

Pattern Type	d_t	Code
Natural	0.0	1 0 0 0 0
Upward shift	$+0.5, +1.5, +2.5, +3.5$	0 1 0 0 0
Downward shift	$-0.5, -1.5, -2.5, -3.5$	0 0 1 0 0
Upward trend	$+0.1, +0.2, +0.3, +0.4$	0 0 0 1 0
Downward trend	$-0.1, -0.2, -0.3, -0.4$	0 0 0 0 1

significant factors of operation for the neuronal network. The results are given in section 5.1 and in all cases, there were no violations to the normality and independence of the residuals $e_{i,j}$.

Table 2: Experimental Design



CONCEPT		DESCRIPTION	
Type of experiment		2^k, fraction 1/16	
Number of factor		7	
Resolution		III	
Number of replicates		3	
Generating process of Alias		D = AB, E = AC, F = BC and G = ABC	
Independent variable (response)		Efficiency of the neural network (η) = correct predictions/total vectors.	
Significance level		0.1	
FACTORS		LEVELS	
DESCRIPTION		LOW	HIGH
A	ρ_{a1}	0.2	0.8
B	ρ_{ab1}	0.2	0.8
C	β_1	0.2	1.0
D	$\rho_{b1,2}$	0.2	0.8
E	ρ_{ab2}	0.2	0.8
F	β_2	0.2	1.0
G	ρ_{a2}	0.2	0.8

5.1 Experimental results

The results lead to the identification of the factors that most influence the prediction efficiency of the neuronal network (characterization). Results also showed the determination of the best combination for the factor levels (relative optimization) that generated higher efficiencies. It was observed that at the level of significance of the experimental test, all the factors influence in the efficiency of the neural network. The relative optimal levels are shown in table 3.

The efficiency of the neural network (η) considering the factors and levels indicated in table 3 are given in table 4 and figure 7. The experimental test validation was carried out with 17,000 data samples created by simulation.

Table 3: ANN Factors and optimal levels

Factors		Optimal Levels
A	(ρ_{a1})	0.8
B	(ρ_{ab1})	0.2
C	(β_1)	1.0
D	$(\rho_{b1,2})$	0.8
E	(ρ_{ab2})	0.2
F	(β_2)	0.2
G	(ρ_{a2})	0.6

Table 4: Neural network efficiency (η)

Pattern Data	η
Natural	30 %
Upward shift	93.5 %
Downward shift	96.4 %
Upward trend	99.8 %
Downward trend	96.4 %

6 Analysis of pattern variation

There is a direct relationship between the value d_t assumed by each special pattern data (table 1), the value of the standard deviation (σ) of the sample data and the efficiency of the neuronal network (η). table 5, shows this situation. In all cases, while the absolute value of d_t increases the corresponding σ and η also increases. A polynomial regression analysis was carried out (σ and d_t are the independent variables and η the dependent variable). The coefficient of correlation was high and typically above 89% (see table 6). This fact indicates that as d_t approaches n_t and σ is low then the neural network may predict a wrong pattern. On the opposite, with higher values of d_t and σ the neural network prediction efficiency increases.

Table 5: Relationship between d_t, standard deviation (σ) and the neural network efficiency (η).

Pattern Type	**Parameters**	**Values**			
Upward shift	η	76.0%	86.8%	92.7%	93.5%
Downward shift	η	42.4%	77.6%	88.2%	96.4%
	d_t	±0.5	±1.5	±2.5	±3.5
	σ	1.009	1.237	1.620	2.089
Upward trend	η	22.0%	25.0%	73.8%	99.8%
Downward trend	η	79.1%	82.8%	95.8%	96.4%
	d_t	±0.1	±0.2	±0.3	±0.4
	σ	0.539	0.670	0.862	1.080

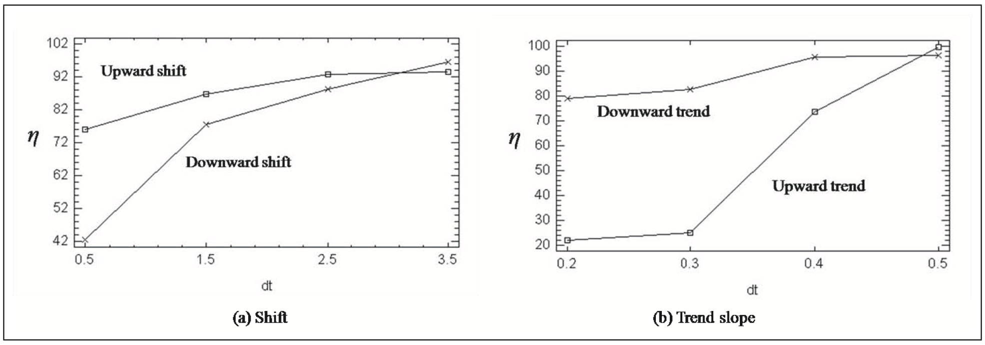


Figure 7: Neural network efficiency (η) vs d_t

Table 6: Equations for the multiple linear regressions

Pattern Type	Multiple Regression	R^2
Upward shift	Efficiency = $107.7+21.4(d_t)-42.5(\sigma)$	99.8
Downward shift	Efficiency = $132.2-61.0(d_t)-119.7(\sigma)$	99.3
Upward trend	Efficiency = $-121.5-294.1(d_t)+317.5(\sigma)$	96.0
Downward trend	Efficiency = $77.3-92.3(d_t)-15.0(\sigma)$	89.21

7 Conclusions

This investigation confirms the obtained results from previous studies with respect to the efficiency of the neural network in the recognition of statistical pattern data. It is also demonstrated, that as the effect of the special cause approaches close to zero, the efficiency decreases because the standard deviation of these data is smaller or equal to 1, i.e., the standard deviation of a data set from a natural pattern. Another result from this investigation is the definition of the most appropriate parameter values for the FuzzyARTMAP that facilitated the use of this neuronal network. It is important to mention that in this application the efficiency of the network for the control chart pattern recognition depends greatly on the d_t and σ values.

Future work has been envisaged to look at the dt and σ values in the confusion zone and their relationship with the sampling window size in order to analyse the neural network behaviour in this zone.

Acknowledgements

The authors wish to thank CIATEC, A.C., and the Posgrado Interinstitucional en Ciencia y Tecnologia (PICyT) for the facilities provided to J. A. Vazquez-Lopez during his doctoral studies.

Bibliography

[1] D.C. Montgomery. *Introduction to Statistical Quality Control*. Third edition. John Wiley & Sons, New York, 1991.

[2] R.S. Guh. Real-time pattern recognition in statistical process control: a hybrid neural network/decision tree-based approach. *IMechE, Part B: J. Engineering Manufacture*. Vol. 219, No. 3. pp. 283-298, 2005.

[3] T. L. Lucy-Bouler. Application to forecasting of neural network recognition of shifts and trends in quality control data. *Proceedings of the World Congress on Neural Networks (WCNN'93)*. Vol. 1, pp. 631-633, UK, 1993.

[4] R.S. Guh. Robustness of the neural network based control chart pattern recognition system to non-normality. *Int. J. Qual. Reliability Mgmt*. Vol. 19(1), pp. 97-112, 2002.

[5] C.W. Zobel, D.F. Cook, Q.J. Nottingham. An augmented neural network classification approach to detecting mean shifts in correlated manufacturing process parameters. *Int. Journal of Production Research*. Vol. 42 (4), pp. 741-758, 2004.

[6] F. Zorriassantine, J.D.T. Tannock. A review of neural networks for statistical process control. *Journal of Intelligent Manufacturing*. Vol. 9, pp. 209-224, 1998.

[7] C.S. Cheng. A multi-layer neural network model for detecting changes in the process mean. *Computers Ind. Engng*. Vol. 28(1), pp. 51-61, 1995.

[8] H.B. Hwarng, and C. W. Chong. Detecting process nonrandomness through a fast and cumulative learning ART-based pattern recognizer. *Int. J. Prod. Res.*, Vol. 33(7), pp. 1817-1833, 1995.

[9] C.S. Cheng. A neural network approach for the analysis of control chart patterns. *Int. J. Prod. Res.*, Vol. 35(3), pp. 667-697, 1997.

[10] E.S. Ho and S. I. Chang. An integrated neural network approach for simultaneous monitoring of process mean and variance shifts - a comparative study. *Int. J. Prod. Res.*, Vol. 37(8), pp. 1881-1901, 1999.

[11] R.S. Guh and J.D.T. Tannock. Recognition of control chart concurrent patterns using a neural network approach. *Int. J. Prod. Res.*, Vol. 37(8), pp. 1743-1765, 1999.

[12] M.A. Wani, and D.T. Pham. Efficient control chart pattern recognition through synergistic and distributed artificial neural networks. *Proc. Instn Mech. Engrs, Part B: J. Engineering Manufacture*. Vol. 213(B2), pp. 157-169, 1999.

[13] D.F Cook, C.W. Zobel and M. L. Wolfe Environmental statistical process control using an augmented neural network classification approach. *European Journal of Operational Research*. Vol. 174, pp. 1631-1642, 2006.

[14] J.A. Swift. Development of a Knowledge-Based Expert System for Control Chart Pattem Recognition and Analysis. *Doctoral Thesis*. Oklahoma State University, 1987.

[15] Gerson Tontini. Pattern Identification in Statistical Process Control Using Fuzzy Neural Networks. *IEEE Management Sciences*. Department Regional University of Blumenau (FURB), 1996.

[16] G.A Carpenter, S. Grossberg, N. Markuzon, J.H. Reynolds, and D.B. Rosen. Fuzzy ARTMAP: A neural network architecture for incremental learning of analog multidimensional maps. *IEEE Transactions on Neural Networks*. Vol. 3 (5), pp. 698-713, 1992.

[17] G. A. Carpenter, S. Grossberg. Adaptive Resonance Theory (ART). *The Hanbook of Brain Theory and Neural Networks*. Edited by M. A. Arbib, The MIT Press. pp. 79-82, 1995.

[18] G. A. Carpenter, S. Grossberg. A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine. *Computer Vision, Graphics, and Image Processing*. Academic Press, Inc. pp. 54-115, 1987.

[19] G. A. Carpenter, S. Grossberg, D. B. Rosen. ART 2-A: An Adaptive Resonance Algorithm for Rapid Category Learning and Recognition. *Neural Networks*. Vol. 4, pp. 493-504, 1991.

[20] G. A. Carpenter, S. Grossberg, J. H. Reynolds. ARTMAP: Supervised Real-Time Learning and Classification of Nonstationary Data by Self-Organizing Neural Network. *Neural Networks*. pp. 565-588, 1991.

[21] J. R. Williamson. Gaussian ARTMAP: A Neural Network for Fast Incremental Learning of Noisy Multidimensional Maps. *Neural Networks*. Vol. 9, No. 5, pp. 881-897, 1996.

[22] G. A. Carpenter, W. D. Ross. ART-EMAP: A Neural Network Architecture for Object Recognition by Evidence Accumulation. *IEEE Trans. on Neural Networks*. Vol. 6, No. 4, pp. 805-818, 1995.

[23] D. Psichogios, L. Ungar. A Hybrid Neural Network-First Principles Approach to Process Modeling. *Computers & Chemical Engineering*. Vol. 38(10), pp. 1499-1511, 1992.

[24] G. Acuña, and E. Pinto. Development of a Matlab®Toolbox for the Design of Grey-Box Neural Models. *Int. Journal of Computers, Communications & Control*. Vol. I, No. 2, pp. 7-14, 2006.

[25] Imre J. Rudas, János Fodor. Intelligent Systems. *Int. Journal of Computers, Communications & Control*, 3(S):132-138, 2008.

Jose Antonio Vazquez-Lopez was born in Celaya, Guanajuato, Mexico in 1972. He graduated with a BEng and MSc in Industrial Engineering from Instituto Tecnologico de Celaya in 1999 and 2001 respectively. He obtained his PhD from the Posgrado Interinstitucional en Ciencia y Tecnologia (PICyT) in 2009. Currently, he is a Lecturer at The Instituto Tecnologico de Celaya. His main research interest is in statistical industrial engineering, where he has also served as industrial consultant for several companies.

Ismael Lopez-Juarez was born in Tequixquiac, Mexico in 1968. He obtained a BEng from The National Autonomous University of Mexico (UNAM) in 1991. He obtained an MSc in Instrument Design and Application at The University of Manchester in 1996 and a PhD in Intelligent Robotics at The Nottingham Trent University in 2000, both in th U.K. He was the founder and leader of the Mechatronics and Intelligent Manufacturing Systems Research Group at CIATEQ, A.C. during 2000-2006. Currently, he is a Principal Researcher at the Robotics and Advanced Manufacturing Systems Research Group at CINVESTAV-Saltillo. Dr. Lopez-Juarez has published over one hundred papers in refereed international journals and conference proceedings in the area of Instrumentation, Self-adaptive Industrial Robotics and Neural Networks.

Mario Peña-Cabrera was born in Queretaro, Mexico in 1954. He graduated with a BEng in Mechanical and Electrical Engineering with specialization in control, communications and electronics from The National Autonomous University of Mexico (UNAM). He holds a MEng in Electronics from the same University and a MEng from The McMaster University, Canada. He obtained his PhD degree in Science and Technology from PICyT, SEP-CONACYT, Mexico. His areas of interest are in robot vision, automation and digital control, areas in which he has been working for over 30 years. Currently, He is the Head of the Computing Engineering Science and Automation Department at The Institute of Research on Applied Mathematics and Systems in The National Autonomous University of Mexico (UNAM).

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 2, pp. 216-223

Synthetic Genes for Artificial Ants. Diversity in Ant Colony Optimization Algorithms

S.C. Negulescu, I. Dzitac, A.E. Lascu

Sorin C. Negulescu
"Lucian Blaga" University of Sibiu, "Hermann Oberth" Faculty of Engineering
10, Victoriei Bd, 550024 Sibiu, Romania
Email: sorin.negulescu@ulbsibiu.ro

Ioan Dzitac
"Aurel Vlaicu" University of Arad, Faculty of Exact Sciences,
Department of Mathematics-Informatics,
Str. Elena Dragoi, Nr. 2, Complex Universitar M, Arad, Romania
and
R&D Agora Ltd. Oradea [Cercetare Dezvoltare Agora]
8, Piata Tineretului, 410526 Oradea, Romania
E-mail ioan.dzitac@uav.ro

Alina E. Lascu
"Lucian Blaga" University of Sibiu, "Hermann Oberth" Faculty of Engineering
10, Victoriei Bd, 550024 Sibiu, Romania
E-mail alina.lascu@ulbsibiu.ro

Abstract: Inspired from the fact that the real world ants from within a colony are not clones (although they may look alike, they are different from one another), in this paper, the authors are presenting an adapted ant colony optimisation (ACO) algorithm that incorporates methods and ideas from genetic algorithms (GA). Following the first (introductory) section of the paper is presented the *history* and the *state of the art*, beginning with the stigmergy and genetic concepts and ending with the latest ACO algorithm variants as multiagent systems (MAS). The *rationale* and the *approach* sections are aiming at presenting the problems with current stigmergy-based algorithms and at proposing a (possible - yet to be fully verified) solution to some of the problems ("synthetic genes" for artificial ants). A *model* used for validating the proposed solution is presented in the next section together with some preliminary simulation results. Some of the conclusions regarding the main subject of the paper (synthetic genes: agents within the MAS with different behaviours) that are closing the paper are: a) the convergence speed of the ACO algorithms can be improved using this approach; b) these "synthetic genes" can be easily implemented (as local variables or properties of the agents); c) the MAS is self-adapting to the specific problem that needs to be optimized.

Keywords: Ant Colony Optimization, Genetic Algorithms, Multiagent Systems, Stigmergy.

1 Introduction

There are about ten thousand species of ants living on earth, grouped in colonies, all of them having or one or (at most) two queens and sterile female workers (all the other ants). As the research conducted by biologists showed [5], there is no central control or no management whatsoever inside a colony, but the behaviour of the colony changes (as it gets older and larger). It was also determined that there are four types of ants [5]: *harvester* ants, *foraging* ants (out on a foraging trail searching for food or bringing food back), *patrolling* (out early in the morning before the foragers are active, somehow choosing the directions that the forager ants will go, and just by coming back to the nest they indicate to the foragers that is safe to go out), *maintenance* (work inside the nest with moist soil for building the walls), *midden work* (put some kind of territorial chemical in the garbage they create). Another interesting fact is that 50% of the ants are doing nothing, acting as a reserve for exceptionally situations.

One class of multi-agent systems (MAS) are copying some of the characteristics of the *foraging ants* within the colonies (depending on how much these characteristics are helpful in solving a particular type of problem) such as [10]:

- being a distributed society of autonomous individuals/agents;
- having fully distributed control among the agents;
- having localized communications among the individual;
- taking stochastic decisions;
- having system-level behaviours that are transcending the behavioural repertoire of the single (minimalist) agent;
- following simple interaction rules.

As a result, the overall very important features of these systems are: *robustness, adaptability and scalability*.

The rest of this paper is arranged as follows: the *history* section presents the stigmergy and genetic concepts and variants of ACO algorithms as MAS. The *rationale* and the *approach* sections are aiming at presenting the problems with current stigmergy-based algorithms and at proposing a (possible - yet to be fully verified) solution to some of the problems ("synthetic genes" for artificial ants). A *model* used for validating the proposed solution is presented in the next section together with some preliminary simulation results. Some of the conclusions regarding the main subject of the paper presented in the last section are: a) the convergence speed of the ACO algorithms can be improved using this approach; b) these "synthetic genes" can be easily implemented (as local variables or properties of the agents); c) the MAS is self-adapting to the specific problem that needs to be optimized.

2 History and related work

Swarm intelligence systems (inspired from biological systems as ant colonies, bird flocking, animal herding, bacterial growth, and fish schooling) are in fact simple agents that are interrelated, being able to communicate one with another and to interact with their environment. They follow simple rules and there is no centralized control [9]. In swarm intelligence systems the focus is not on modelling, inspired from biologic (sub-symbolic) paradigms, but merely on simulating biologic behaviour [8].

The ant colony optimization algorithm is a probabilistic technique for solving computational problems which can be reduced to finding good paths through graphs [17]. This technique was initially proposed by Marco Dorigo in 1992 in his PhD thesis [6] [7] and it was aiming at searching an optimal path in a graph (i.e. the travelling salesperson problem - TSP) based on the behaviour of ants seeking a path between their colony and a source of food [4].

The algorithm was further developed and, as a result, some common extensions appeared such as the *Elitist Ant System* (EAS) where the ant that found a better solution could deposit an extra amount of pheromone; the *Max-Min Ant System* (MMAS)[16] where the pheromone intensity was bounded to certain minimum and maximum values (Tmax, Tmin), and where only the ant that found a better solution could deposit pheromone. The MMAS was the first algorithm that was constantly checking its convergence to solution; all the edges in the graph were initialized to Tmax and reinitialized to Tmax when the algorithm was near stagnation [17].

Another version of the ACO is the *Rank-Based Ant System* (ASrank) where all solutions are ranked according to their fitness. The amount of pheromone deposited is then weighted for each solution, such that the solutions with better fitness deposit more pheromone than the solutions with worse fitness [17].

Besides these common variants of the algorithm [12], the authors proposed some different approaches such as:

- **Human-Driven Stigmergic Control** [3] - aimed at exploring the relationship between *stigmergy* and *synergy*, based on the *threshold principle*, specifically, focusing on affordability and keeping a definite engineering perspective, the purpose was to save computer resources in applying stigmergic control to industrial problems by exploring the relationship between the number of digital ants and problem complexity. The long-range target was to follow the analogy to superconductivity: *moving the threshold* in order to improve performance and/or save computing resources. The research proved that the threshold exists and it depends on problem type and complexity; the same solution quality can be obtained with fewer ants.
- **User-Driven Heuristics** [1] [12] - the approach emphasised the significance of the environment in the agent system and the attention that must be paid to the dynamics that emerges from the indirect interactions of the agents. Finding out the form and parameters that influence the system behaviour is non-trivial for humans and the research investigated the methods that would enable user-driven solutions in dealing with system tuning. The proposed methods were based on two essential functional requirements: a) at the macro level, the need to monitor and represent in an intuitive way the system behaviour and b) at the micro level, the need to control and track the system state space. The approach proved to be workable on usual configurations and effective in dealing with combinatorial explosion.

- **(Sub-)Symbolic Inferences in Multi-Agent Systems** [2] - the research aimed at showing that new logics, already useful in modern software engineering, become necessary mainly for MAS, despite obvious adversities. The features asked for by the paradigm of *computing as intelligent interaction*, based on "nuances of nuanced-reasoning", that should be reflected by agent logics were outlined. By *injecting* symbolic reasoning in systems based on sub-symbolic "emergent synthesis" the authors presented the way in which how quantifiable synergy can be reached - even in advanced challenging domains, such as stigmergic coordination.

The original idea has diversified to solve a wider class of numerical problems, and as a result, several problems have emerged, drawing on various aspects of the behaviour of ants such as solving combinatorial optimization, classification and image processing problems [15].

The authors successfully applied the ACO algorithms to solve difficult problems such as: *vehicle route allocation* (with multiple constraints) [13] [14] and *optimal capacitor banks placement in power distribution networks* [15] where the criterion of the mathematical optimization model was a nonlinear function based on costs and the model imposed equality constraints described by the network operating equations and inequality constraints required to maintain within admissible limits the parameters characterizing the system state.

3 Rationale and approach

The motive for proposing synthetic genes for artificial ants and promoting the diversity in ant colony optimization algorithms is that the convergence speed of current variations of ACO algorithms is *strongly dependent of the problem type and size*. The real world ants from within a colony are not clones as current ACO algorithms are modelling them; although they may look alike, they are different from one another.

The vast majority of ACO algorithms have general variables that are determining the overall system behaviour such as:

- pheromone evaporation speed when updating the pheromones intensities in the graph (ρ);
- pheromone quantity deposited on graph (Q);
- pheromone influence is on edge selection (α);
- distance influence on edge selection (β);
- the multiplication value for the pheromone intensity of the elitist ant (ε) - only for EAS.

When a certain problem must be solved, the values of the above presented parameters are set and after a few runs they are adjusted in order to improve the overall convergence speed or solution quality.

When dealing with dynamic optimisation problems the above mentioned approach accentuates once more its weak points, as it is almost impossible to determine good parameter values.

A possible manner to combat these limitations is to adopt some ideas from the GA field. Although hybrid approaches between MAS (such as ACO or EAS) and GA are not new, the technique proposed herein is innovative because:

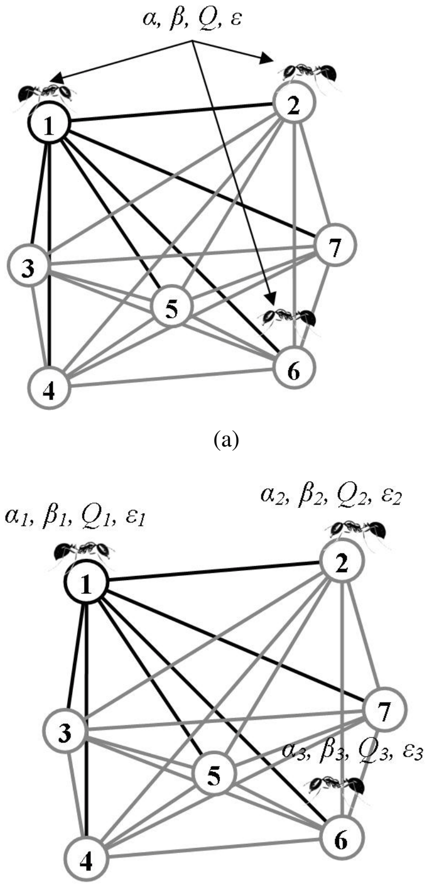
- it does not absolutely require to implement all the GA concepts (like selection, crossover and mutation);
- the fitness function may or may not exist (depending on problem type);
- the generations from GA can be associated to the iteration number from MAS;
- the global parameters belonging to the algorithm are converted to local parameters belonging to ants.

As depicted in figure 1b, each virtual ant within the colony has its own governing parameters in the same manner that real ants exhibit different behaviour. More details about the way in which such a system should work are presented in the next section.

4 The model

Since the model proposed herein imposes that each virtual ant within the colony must have its own parameters that are determining its behaviour, it is important for the user to have the ability to specify some general limits in which these parameters may vary. In this regard, some other general parameters must be defined as follows:

- minimum and maximum values for the *exploitation* of the good old found paths - α_{min}, α_{max};
- minimum and maximum values for the *exploration* of the other paths within the graph - β_{min}, β_{max};
- minimum and maximum values for the multiplication value for the pheromone intensity of the (elitist) ants that found a better solution - ε_{max}, ε_{max};



(a)

(b)

Figure 1: Exemplification of a) global parameters versus b) local parameters for the ant system

Depending on the problem to be solved some other global parameters can be defined to enforce the limits for the local parameters (e.g. for the vehicle route allocation problem [13] a local history parameter was used and H_{min}, H_{max} as global parameters).

After finding a better solution, different actions can be taken by the algorithm to auto-tune the ant local parameters:

• the parameters will be adjusted for *some*, *all* or *none* of the ants;

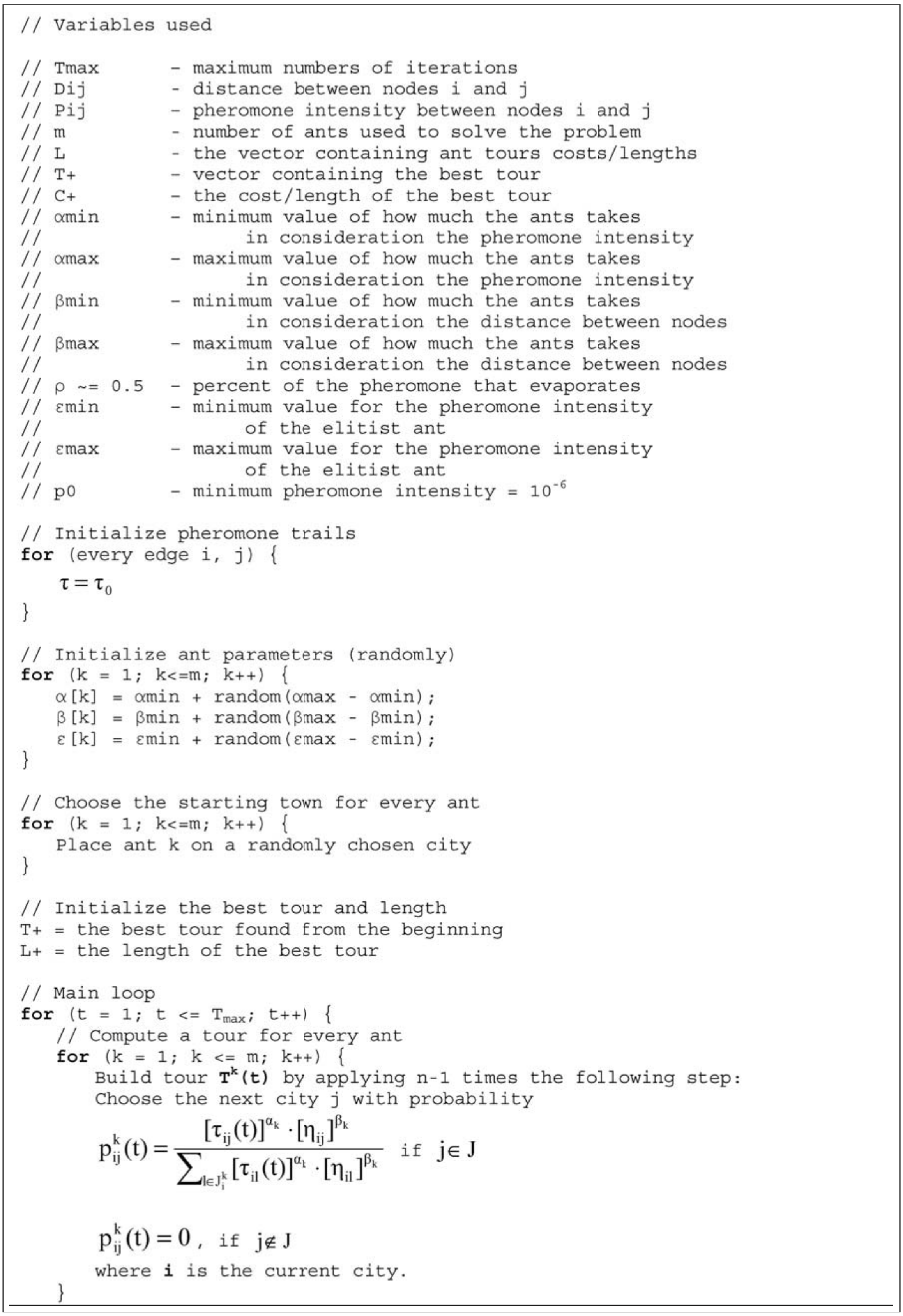
• parameters can be adjusted using different types of equations - e.g. form simple average, weighted average (depending on solution quality), or other linear functions to polynomial or exponential functions.

The initialization of the ant parameters can be easily done in a random manner as shown in figure 2, but some other initialization methods can be applied as well (e.g. Gaussian centred to the average value: min+(max-min)/2).

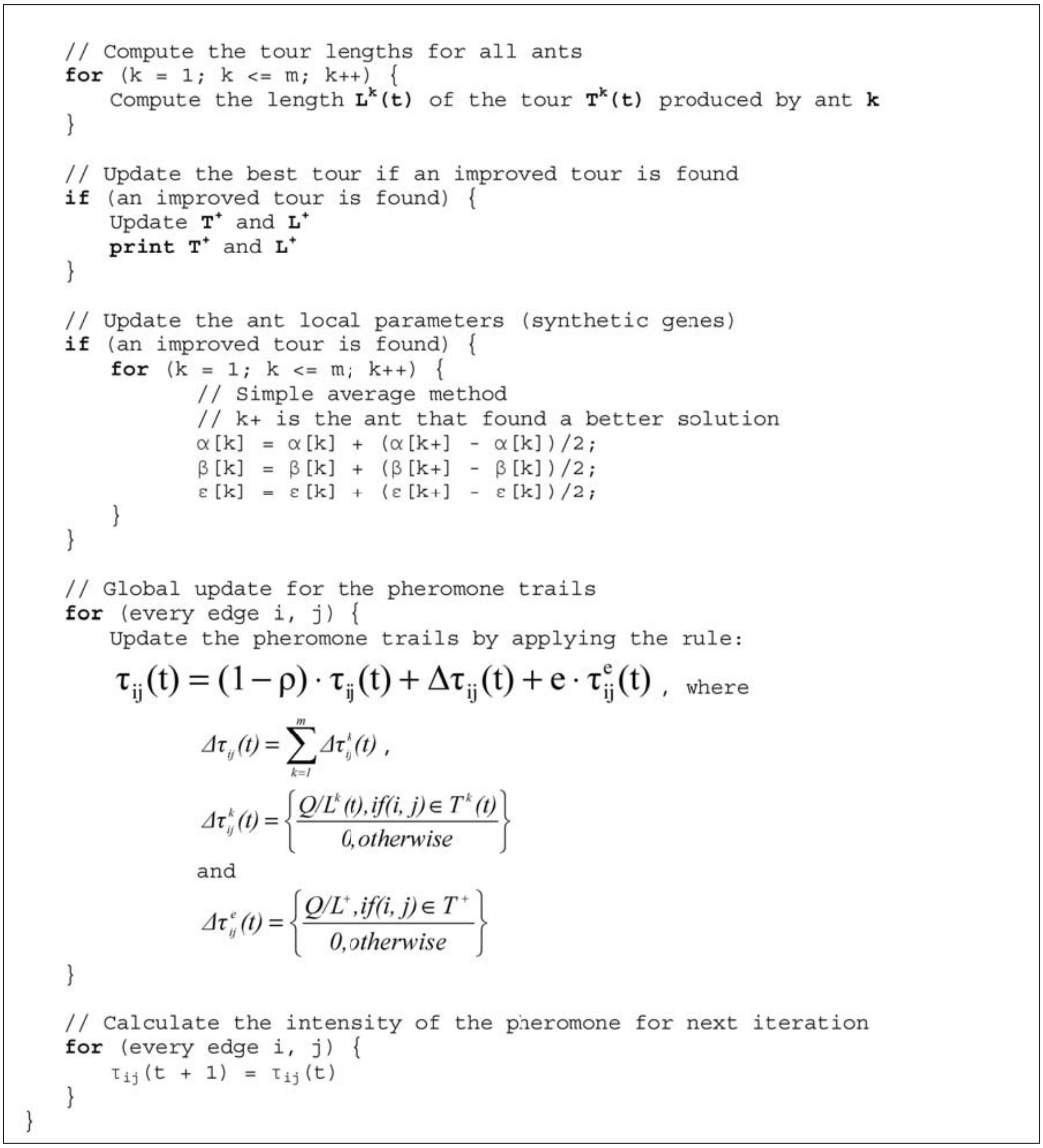
Although the research on the proposed model is in progress the results so far are promising. The algorithm is self-tuning to the problem type and size. Some other methods must be checked such as starting with a big number of ants and reducing it to a minimum in time, with a twofold advantage:

• having a large genetic pool at the beginning will assure that the algorithm can adapt to very diverse problems;

• reducing the number of ants will increase the computation speed; this number can be drastically reduced as presented in [3].



```
// Variables used

// Tmax        - maximum numbers of iterations
// Dij         - distance between nodes i and j
// Pij         - pheromone intensity between nodes i and j
// m           - number of ants used to solve the problem
// L           - the vector containing ant tours costs/lengths
// T+          - vector containing the best tour
// C+          - the cost/length of the best tour
// αmin        - minimum value of how much the ants takes
//                    in consideration the pheromone intensity
// αmax        - maximum value of how much the ants takes
//                    in consideration the pheromone intensity
// βmin        - minimum value of how much the ants takes
//                    in consideration the distance between nodes
// βmax        - maximum value of how much the ants takes
//                    in consideration the distance between nodes
// ρ ~= 0.5    - percent of the pheromone that evaporates
// εmin        - minimum value for the pheromone intensity
//                    of the elitist ant
// εmax        - maximum value for the pheromone intensity
//                    of the elitist ant
// p0          - minimum pheromone intensity = $10^{-6}$

// Initialize pheromone trails
for (every edge i, j) {
    $\tau = \tau_0$
}

// Initialize ant parameters (randomly)
for (k = 1; k<=m; k++) {
    α[k] = αmin + random(αmax - αmin);
    β[k] = βmin + random(βmax - βmin);
    ε[k] = εmin + random(εmax - εmin);
}

// Choose the starting town for every ant
for (k = 1; k<=m; k++) {
    Place ant k on a randomly chosen city
}

// Initialize the best tour and length
T+ = the best tour found from the beginning
L+ = the length of the best tour

// Main loop
for (t = 1; t <= $T_{max}$; t++) {
    // Compute a tour for every ant
    for (k = 1; k <= m; k++) {
        Build tour $T^k(t)$ by applying n-1 times the following step:
        Choose the next city j with probability
        $p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^{\alpha_k} \cdot [\eta_{ij}]^{\beta_k}}{\sum_{l \in J_i^k} [\tau_{il}(t)]^{\alpha_k} \cdot [\eta_{il}]^{\beta_k}}$ if $j \in J$

        $p_{ij}^k(t) = 0$, if $j \notin J$
        where i is the current city.
    }
```



```
    // Compute the tour lengths for all ants
    for (k = 1; k <= m; k++) {
        Compute the length $L^k(t)$ of the tour $T^k(t)$ produced by ant k
    }

    // Update the best tour if an improved tour is found
    if (an improved tour is found) {
        Update $T^+$ and $L^+$
        print $T^+$ and $L^+$
    }

    // Update the ant local parameters (synthetic genes)
    if (an improved tour is found) {
        for (k = 1; k <= m; k++) {
                // Simple average method
                // k+ is the ant that found a better solution
                α[k] = α[k] + (α[k+] - α[k])/2;
                β[k] = β[k] + (β[k+] - β[k])/2;
                ε[k] = ε[k] + (ε[k+] - ε[k])/2;
        }
    }

    // Global update for the pheromone trails
    for (every edge i, j) {
        Update the pheromone trails by applying the rule:
        $\tau_{ij}(t) = (1-\rho)\cdot\tau_{ij}(t) + \Delta\tau_{ij}(t) + e\cdot\tau^{e}_{ij}(t)$ , where
                $\Delta\tau_{ij}(t) = \sum_{k=1}^{m}\Delta\tau^{k}_{ij}(t)$ ,
                $\Delta\tau^{k}_{ij}(t) = \left\{\frac{Q/L^{k}(t), if(i,j)\in T^{k}(t)}{0, otherwise}\right\}$
                and
                $\Delta\tau^{e}_{ij}(t) = \left\{\frac{Q/L^{+}, if(i,j)\in T^{+}}{0, otherwise}\right\}$
    }

    // Calculate the intensity of the pheromone for next iteration
    for (every edge i, j) {
        $\tau_{ij}(t + 1) = \tau_{ij}(t)$
    }
}
```

Figure 2: The proposed pseudocode for the "synthetic genes" enhanced EAS.

5 Conclusions and future work

Since this paper presents an ongoing research, the conclusions will be presented tanking in consideration two time ranges: a) *short* (current relevant results) and b) *medium-long* (directions of future work).

a1. The proposed ACO algorithms have the ability to self-adapt to different optimisation problem types and sizes.

a2. The convergence speed of the ACO algorithms can be improved using this approach.

a3. The "synthetic genes" can be easily implemented (as local variables or properties of the agents); only some other global variables that will impose the limits of variation must be defined.

a4. Different types of initialisation of the genes values can be used (e.g. random). Ideas for this particular subject can be used from the GA field.

a5. The genes values can be automatically adjusted for *some*, *all* or *none* of the ants. Moreover, this can be achieved using different types of equations - e.g. form simple average, weighted average (depending on solution quality), or other linear functions to polynomial or exponential functions.

b1. Some other methods of initialization/algorithm evolution must be tested. As presented in the previous section, the algorithm can start with a big number of ants and reduce it to a minimum in time.

b2. Other concepts form GA can be implemented taking into account both the advantages (solution quality) and the disadvantages (computing time) like *selection*, *crossover* or *mutation*.

b3. Finding a solution to the stagnation problem that is intrinsic both to ACO and GA approaches. Another possible technique would be to reinitialize the local parameters while keeping the pheromones intensities on the graph intact.

Bibliography

[1] Bărbat B.E, Moiceanu A., Pleşca S., Negulescu S.C., Affordability and Paradigms in Agent-Based Systems, *Computer Sc. J. of Moldova*, 15, 2(44):178-201, 2007.

[2] Bărbat B.E., Negulescu S.C., From Algorithms to (Sub-)Symbolic Inferences in Multi-Agent Systems. *International Journal of Computers, Communications and Control*, 1(3):5-12, 2006.

[3] Bărbat B.E., Negulescu S.C., Zamfirescu C.B., Human-Driven Stigmergic Control. Moving the Threshold. In N. Simonov (Ed.), *Proc. of the 17th IMACS World Congress (Scientific Computation, Applied Mathematics and Simulation)*, Paris, pp.86-92, 2005.

[4] Bonabeau E., Dorigo M., Theraulaz G., *Swarm Intelligence: From Natural to Artificial Systems*, New York: Oxford University Press, 1999.

[5] Deborah Gordon (2008), Dig ants, Retrived 11.2009, from TED. Ideas worth spreading. Web site: http://www.ted.com/talks/lang/eng/deborah_gordon_digs _ants.html.

[6] Dorigo M., Maniezzo V., Colorni A., The Ant System: Optimisation by a Colony of Cooperating Agents, *IEEE Transactions on Systems, Man, and Cybernetics*, Part B, 26 (1):29-42, 1996.

[7] Dorigo, M., *Optimization, Learning and Natural Algorithms*, PhD thesis. Italy: Politecnico di Milano, 1992.

[8] Dzitac I., Bărbat B.E., Artificial Intelligence + Distributed Systems = Agents, *International Journal of Computers, Communications and Control*, 4(1):17-26, 2009.

[9] Dzitac I., Moisil I., Advanced AI Techniques for Web Mining, *Proceedings of the 10th WSEAS international conference on Mathematical methods, computational techniques and intelligent systems*, Corfu, pp.343-346, 2008.

[10] Gambardella L.M., Di Caro G. (2005), The Ant Colony Optimization (ACO) Metaheuristic: a Swarm Intelligence Framework for Complex Optimization Tasks. Retrived 2008, from University of Bologna: First Summer School on Aspects of Complexity. Web site: http://www.cs.unibo.it/ fioretti/AC/AC2005/ docs/slides_dicaro.pdf.

[11] Negulescu S.C., Bărbat B.E., Enhancing the effectiveness of simple multi-agent systems through stigmergic coordination, In ICSC-NAISO (Ed.), *Fourth International ICSC Symposium on Engineering Of Intelligent Systems* (EIS 2004), Canada: ICSC-NAISO Academic Press, pp.149-156, 2004.

[12] Negulescu S.C., Zamfirescu C.B., Bărbat B.E., User-Driven Heuristics for nondeterministic problems. *Studies in Informatics and Control* (Special issue dedicated to the 2nd Romanian-Hungarian Joint Symp. on Applied Computational Intelligence), 15(3):289-296, 2006.

[13] Negulescu, S.C., Kifor, C.V., Oprean, C., Ant Colony Solving Multiple Constraints Problem: Vehicle Route Allocation. *International Journal of Computers, Communications and Control* (IJCCC), 3(4):366-373, 2008.

[14] Negulescu, S.C., Oprean, C., Kifor, C.V., Carabulea, I., Elitist ant system for route allocation problem. In Mastorakis, N.E. et al (Ed.), *Proceedings of the 8th conference on Applied Informatics and Communications*, Greece: World Scientific and Engineering Academy and Society (WSEAS), pp.62-67, 2008.

[15] Secui, D.C., Dzitac, S., Bendea, G.V., Dzitac, I., An ACO Algorithm for Optimal Capacitor Banks Placement in Power Distribution Networks. *Studies in Informatics and Control*, 18(4):305-314, 2009.

[16] Stützle T., Hoos H.H., MAX-MIN Ant System, *Future Generation Computer Systems*, 16 (8):889-914, 2000.

[17] Wikipedia, the free encyclopedia (2009), Ant colony optimization. Retrived 11.2009, from WIkipedia. Web site: http://en.wikipedia.org/wiki/ Ant_colony_optimization.

Sorin C. Negulescu has a M.Sc. degree in Quality Mangement (2005) a M.Sc. degree in Computer Science (2003), "Lucian Blaga" University, Sibiu and a Ph. D in industrial engineering. His research interests include multi-agent systems with applications in manufacturing systems using stigmergic coordination, factory automation and the holonic paradigm.

Ioan Dzitac (b. February 14, 1953) received his M.Sc. in Mathematics (1977) and PhD in Information Sciences (2002) from "Babes-Bolyai" University of Cluj-Napoca. Now he is full professor of informatics at Mathematics-Informatics Department, Faculty of Exact Sciences, "Aurel Vlaicu" University of Arad, Romania and director of Research & Development Agora Ltd. of Oradea, Romania. His current research interests include different aspects of Artificial Intelligence and Distributed Systems. He has (co-)authored 18 books and more than 70 papers, more than 50 de conferences participation, member in International Program Committee of more than 40 conferences and workshops.

Alina E. Lascu has a M.Sc. degree in International Relations and European Studies (2008), is currently in the last year of her master studies in the domain of Computer Science and in the first year of the doctoral studies in the domain of Computers and ICT. As depicted from her educational background her research interests are very much trans-disciplinary: Language Technology, Intelligent Agents, Agent Orientation, Computer-Aided Semiosis, Cultural and Interdisciplinary Studies.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 2, pp. 224-237

Towards Structured Modelling with Hyperdag P Systems

R. Nicolescu, M.J. Dinneen, Y.-B. Kim

Michael J. Dinneen, Yun-Bum Kim and Radu Nicolescu
Department of Computer Science, University of Auckland,
Private Bag 92019, Auckland, New Zealand
E-mail: `{radu,mjd,yun}@cs.auckland.ac.nz`

Abstract: Although P systems are computationally complete, many real world models, such as socio-economic systems, databases, operating systems and distributed systems, seem to require more expressive power than provided by tree structures. Many such systems have a primary tree-like structure augmented with shared or secondary communication channels. Modelling these as tree-based systems, while theoretically possible, is not very appealing, because it typically needs artificial extensions that introduce additional complexities, inexistent in the originals.
In this paper, we propose and define a new model called *hyperdag P systems*, in short, *hP systems*, which extend the definition of conventional P systems, by allowing dags, interpreted as hypergraphs, instead of trees, as models for the membrane structure.
We investigate the relation between our hP systems and neural P systems. Despite using an apparently restricted structure, i.e., a dag instead of a general graph, we argue that hP systems have essentially the same computational power as tissue and neural P systems. We argue that hP systems offer a structured approach to membrane-based modelling that is often closer to the behavior and underlying structure of the modelled objects.

Keywords: hyperdag P systems, tissue and neural P systems, membrane structures.

1 Introduction

P systems provide a distributed computational model, based on the structure and interaction of living cells, introduced by G. Păun in 1998 [10]. The model was initially based on transition rules, but was later expanded into a large family of related models. Essentially, all versions of P systems have a structure consisting of cell-like membranes and a set of rules that govern their evolution over time.

Many of the "classical" versions use a structure, where membranes correspond to nodes in a rooted tree. Such a structure is often visualized as a Venn diagram, where nesting denotes a parent–child relationship. For example, Figure 1 [12] shows the same P system structure with nine membranes, labelled as $1, \ldots, 9$, both as a rooted tree and as a Venn diagram.

More, recently, neural P systems [11], abbreviated as nP systems (also known as tissue P systems [7]), have been introduced, partially to overcome the limitations of the tree model. Essentially, these systems organize their cells in an arbitrary graph. For example, ignoring for the moment the actual contents of cells (states, objects, rules), Figure 2 illustrates the membrane structure of a simple nP system, consisting of three cells, $\sigma_1, \sigma_2, \sigma_3$, where cell σ_1 is the output cell.

A large variety of rules have been used to describe the operational behavior of P systems, such as multiset rewriting rules, communication rules and membrane handling rules. Essentially, transition P systems and nP systems use multiset rewriting rules, P systems with symport/antiport rules operate by communicating immutable objects and P systems with active membranes combine all three rule types. For a comprehensive overview and more details, we refer the reader to [11, 12].

Besides theoretical computer science and biology, P systems have been applied to a variety of other domains, ranging from linguistics [5] to theoretically efficient solutions of NP-complete problems [16],

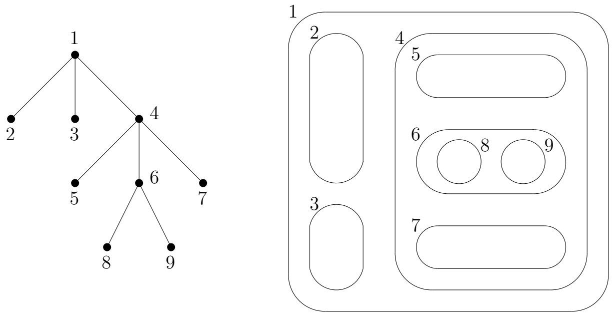


Figure 1: A P system structure represented as a tree and as a Venn diagram.

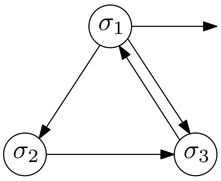


Figure 2: An nP system structure represented as a digraph.

or to model distributed algorithms [3, 6]. The underlying tree structure provides good support for reasoning and formal verification, good potential for efficient implementation on multi-core architectures, and an excellent visualization, very appealing to practitioners.

Although the P systems are computationally complete, many real world models seem to require more expressive power, essentially trees augmented by shared or secondary communication channels. For example, the notion of a processing node having a unique parent is not true for (a) computer networks where a computer could simultaneously be attached to several subnets (e.g., to an Ethernet bus and to a wireless cell), (b) living organisms may be the result of multiple inheritance (e.g., the evolutionary "tree" is not really a tree, because of lateral gene transfer [4]) and (c) socio-economic scenarios where a player is often connected to and influenced by more than one factor [13, 14, 15].

Modelling these as tree-based systems, while theoretically possible, is not very appealing. Simulating shared or secondary channels requires artificial mechanisms that will ripple data up and down the tree, via a common ancestor. This could of course limit the merits of using a formal model. Models based on general graphs, such as nP systems, while allowing any direct communications, also tend to obscure the structures already present in the modelled objects, limiting the advantages that a more structured approach could provide. Verification is more difficult without a clear modularization of concerns, practical parallel implementation could be less efficient, if the locality of reference is not enforced, and visualizations are not very meaningful, unless the primary structure is clearly emphasized.

We do not think that we have to choose between structure and flexibility. We propose a solution that seems to combine both, i.e., flexibility without sacrificing the advantages of a structured approach.

Our main contribution in this paper is to propose a new model for P systems, called *hyperdag P systems*, in short, *hP systems*, that allows more flexible communications than tree-based models, while preserving a strong hierarchical structure. This model, defined in Section 3, (a) extends the tree structure of classical P systems to directed acyclic graphs (dags), (b) augments the operational rules of nP systems with broadcast facilities (via a *go-sibling* transfer tag), and (c) enables dynamical changes of the rewriting modes (e.g., to alternate between determinism and parallelism) and of the transfer modes (e.g., to switch between unicast or broadcast). In contrast, classical P systems, both tree- and graph-based P systems, seem to focus on a statical approach.

We investigate the relation between our hP systems and nP systems. Despite using an apparently restricted structure, we show in Section 4 that our dag-based model has the same computational power as graph-based tissue P systems and neural P systems.

We argue that hP systems offer a structured approach to membrane-based modelling that is often closer to the behavior and underlying structure of the modelled objects. Because our extensions address the membrane topology, not the rules model, they can be applied to a variety of P system flavors, such as systems based on symport/antiport rules.

We support our view with a realistic example (see Examples 11 and 12), inspired from computer networking, modelled as an hP system with a shared communication channel (broadcast channel).

Classical P systems allow a "nice" planar visualization, where the parent–child relationships between membranes are represented by Venn-like diagrams. We show in Section 5 that the extended membrane structure of hP systems can still be visualized by hierarchically nested planar regions.

2 Preliminaries

A (binary) *relation* R over two sets X and Y is a subset of their Cartesian product, $R \subseteq X \times Y$. For $A \subseteq X$ and $B \subseteq Y$, we set $R(A) = \{y \in Y \mid \exists x \in A, (x,y) \in R\}$, $R^{-1}(B) = \{x \in X \mid \exists y \in B, (x,y) \in R\}$.

A *digraph* (directed graph) G is a pair (X,A), where X is a finite set of elements called *nodes* (or *vertices*), and A is a binary relation $A \subseteq X \times X$, of elements called *arcs*. For an arc $(x,y) \in A$, x is a *predecessor* of y and y is a *successor* of x. A length $n-1$ *path* is a sequence of n distinct nodes $x_1,\ldots,x_n$, such that $\{(x_1,x_2),\ldots,(x_{n-1},x_n)\} \subseteq A$. A *cycle* is a path $x_1,\ldots,x_n$, where $n \geq 1$ and $(x_n,x_1) \in A$.

A *dag* (directed acyclic graph) is a digraph (X,A) without cycles. For $x \in X$, $A^{-1}(x) = A^{-1}(\{x\})$ are x's *parents*, $A(x) = A(\{x\})$ are x's *children*, $A(A^{-1}(x)) \backslash \{x\} = A(A^{-1}(\{x\})) \backslash \{x\}$ are x's *siblings* (siblings defines a symmetric relation). A node $x \in X$ is a *source* iff $|A^{-1}(x)| = 0$, and $x \in X$ is a *sink* iff $|A(x)| = 0$. The *height* of a node x is the maximum length of all paths from x to a sink node. An arc (x,y) is *transitive* if there exists a path $x_1,\ldots,x_n$, with $x_1 = x$, $x_n = y$ and $n > 2$. Dags without transitive arcs are here called *canonical*.

A (rooted unordered) *tree* is a dag with exactly one source, called *root*, and all other nodes have exactly one parent (predecessor). Sinks in a tree are also called *leaves*. A *topological order* of a dag is a linear reordering of vertices, in which each vertex x comes before all its children vertices $A(x)$.

Dags and trees are typically represented with parent-child arcs on the top-down axis, i.e., sources (roots) up and sinks (leaves) down. Example dags are shown in Figures 3 and 4.

We consider a variant hypergraph definition, based on multisets, as an extension of the classical definition [1], which is based on sets. A *hypergraph* is here a pair (X,E), where X is a finite set of elements called *nodes* (or *vertices*), and E is a finite *multiset* of subsets of X, i.e., $e \in E \Leftrightarrow e \subseteq X$. By using a multiset of edges, instead of a more conventional set of edges, we introduce an *intensional* element, where two *extensionally* equivalent hyperedges (i.e., hyperedges containing the same nodes) are not necessarily equal. A *graph* is a set based hypergraph, where hyperedges are known as *edges* and contain exactly two nodes. Alternatively, a graph (X,E) can be interpreted as a digraph (X,A), where $A = \{(x,y) \mid \{x,y\} \in E\}$. Hypergraphs (set or multiset based) can be represented by planar diagrams, where hyperedges are represented as regions delimited by images of Jordan curves (simple closed curves) [2].

With the above hypergraph definition, a height 1 dag (X,A) can be interpreted as a hypergraph (X,E), where E is the multiset $E = \{A(x) \mid |A^{-1}(x)| = 0\}$. For example, Figure 3 represents, side by side, the dag $D = (\{a,b,c,d,e,f\},\{(d,a),(d,b),(d,c),(e,b),(e,c),(f,b),(f,c)\})$ and its corresponding hypergraph $H = (\{a,b,c\},\{d,e,f\})$, where $d = \{a,b,c\}, e = \{b,c\}, f = \{b,c\}$. Note that the apparently empty differences of regions are needed in the case of *multiset* based hypergraphs, to support the *intensional* (as opposed to the *extensional*) aspect: here $e \neq f$, despite containing the same nodes, b and c, and neither e nor f is included in d.

Generalizing the above hypergraph definition, a height n *generalized hypergraph* is a system (X,E), recursively built via a sequence of n hypergraphs $(X_1,E_1),\ldots,(X_n,E_n)$ where $X_1 = X$, $X_i \cap E_i = \emptyset, X_{i+1} =$

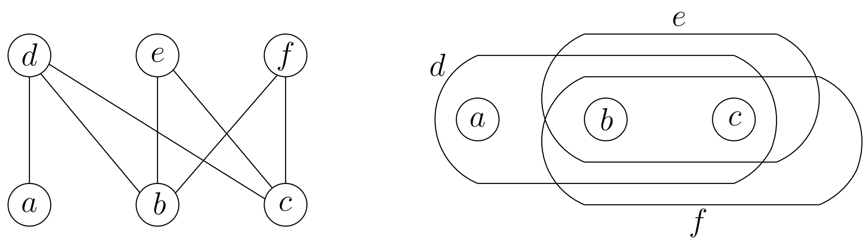


Figure 3: A simple height 1 dag and its corresponding hypergraph representation.

$X_i \cup E_i$, $e \cap E_i \neq \emptyset$ for $\forall e \in E_{i+1}$ and $E = \bigcup_{i \in \{1,...,n\}} E_i$. An arbitrary height n dag can be represented by a height n generalized hypergraph, where the hypergraph nodes correspond to dag sinks, and height i hyperedges correspond to height i dag nodes, for $i \in \{1,...,n\}$.

We will later see that any generalized hypergraph that corresponds to a non-transitive dag can also be represented by hierarchically nested planar regions delimited by Jordan curves, where arcs are represented by direct nesting. For example, Figure 4 shows a height 2 dag and its corresponding height 2 hypergraph (X,E), where $X = X_1 = \{a,b,c,d,e\}$, $E_1 = \{f,g,h\}$, $E_2 = \{i\}$, $E = \{f,g,h,i\}$.

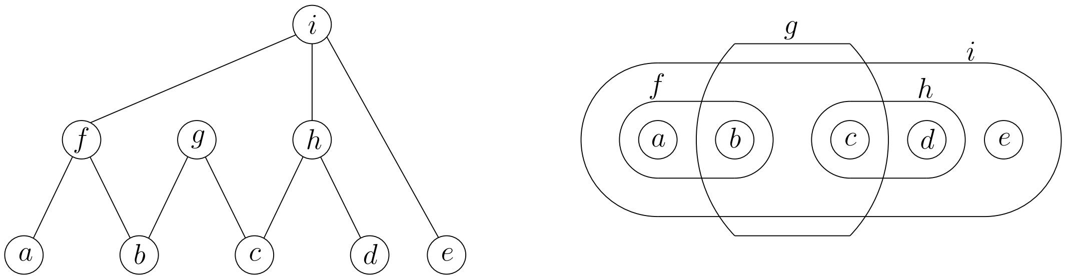


Figure 4: A height 2 dag and its corresponding height 2 hypergraph.

An *alphabet O* is a finite non-empty sets of *objects*. We will assume that the alphabet O is implicitly ordered. *Multisets* over an alphabet O are represented as strings over O, such as $o_1^{n_1} \dots o_k^{n_k}$, where $o_i \in O$, $n_i \geq 0$, and, in the canonical form, letters appear in sorted order, i.e., $o_1 < \cdots < o_k$, and $n_i \geq 1$. The set of all multisets is denoted by O^*. For this representation, two strings are equivalent if they become equal after sorting, e.g., a^2cbd^0a and a^3bc are equivalent representations of the same multiset $\{a,a,a,b,c\}$. Under this convention, the empty string λ represents the empty multiset, and string concatenation represents multiset union, e.g., $(a^2c) \cdot (ab) = a^3bc$.

3 Hyperdag P Systems

In this paper we use the definition of nP systems, as given in [11], that coincides with an early definition of tissue P systems as given in [7]. Our definition includes a small technical correction (slight ambiguity). For details, please see our technical report [8].

As in the mentioned nP systems definition, we will use the following sets of tagged objects: $O_{go} = \{(a,go) \mid a \in O\}$, $O_{out} = \{(a,out) \mid a \in O\}$, and we set $O_{tot} = O \cup O_{go} \cup O_{out}$. For simplicity, we will use subscripts for these tagged objects, such as a_{go} for (a,go) and a_{out} for (a,out). We also define projection homomorphisms, here denoted in postfix notation: $|_O, |_{go}, |_{out} : O_{tot}^* \to O^*$, by $o|_O = o, o_{go}|_{go} = o, o_{out}|_{out} = o$ for $o \in O$, and otherwise λ. For example, $a^2a_{go}^3b^4b_{go}|_{go} = a^3b$.

Besides the existing *go, out* tags, we consider three other object tags:

1. *go-parent*, abbreviated by the symbol $\uparrow$, indicating objects that will be sent to parents;

2. *go-child*, abbreviated by the symbol $\downarrow$, indicating objects that will be sent to children;
3. *go-sibling*, abbreviated by the symbol $\leftrightarrow$, indicating objects that will be sent to siblings.

The precise semantics of these tags will be explained below when we detail the hP system object transfer modes. In fact, we could also discard the *go* tag, as it corresponds to the union of these new target tags (*go-parent, go-child, go-sibling*); however, we will keep it here, for its concise expressive power. We use similar notation as nP systems for these new tags $O_{\uparrow}, O_{\downarrow}, O_{\leftrightarrow}$, and postfix projections $|_{\uparrow}, |_{\downarrow}, |_{\leftrightarrow}$.

Other extension tags, including addressing mechanisms (such as from, to or via tags) are possible, and indeed seem natural, but this is beyond the scope of this article.

We will now define hP systems, as an apparent restriction of nP systems, where the underlying structure is a dag, with several other adjustments.

Definition 1 (Hyperdag P systems) An *hP system* (of degree m) is a system $\Pi = (O, \sigma_1, \ldots, \sigma_m, \delta, I_{out})$, where:

1. O is an ordered finite non-empty alphabet of *objects*;
2. $\sigma_1, \ldots, \sigma_m$ are *cells*, of the form $\sigma_i = (Q_i, s_{i,0}, w_{i,0}, P_i)$, $1 \leq i \leq m$, where:
   - Q_i is a finite set (of *states*),
   - $s_{i,0} \in Q_i$ is the *initial state*,
   - $w_{i,0} \in O^*$ is the *initial multiset* of objects,
   - P_i is a finite set of multiset rewriting *rules* of the form $sx \rightarrow s'x'u_{\uparrow}v_{\downarrow}w_{\leftrightarrow}y_{go}z_{out}$, where $s, s' \in Q_i$, $x, x' \in O^*$, $u_{\uparrow} \in O^*_{\uparrow}$, $v_{\downarrow} \in O^*_{\downarrow}$, $w_{\leftrightarrow} \in O^*_{\leftrightarrow}$, $y_{go} \in O^*_{go}$ and $z_{out} \in O^*_{out}$, with the restriction that $z_{out} = \lambda$, for all $i \in \{1, \ldots, m\} \backslash I_{out}$,
3. δ is a set of dag parent–child arcs on $\{1, \ldots, m\}$, i.e., $\delta \subseteq \{1, \ldots, m\} \times \{1, \ldots, m\}$, representing *duplex* communication channels between cells;
4. $I_{out} \subseteq \{1, \ldots, m\}$ indicates the *output cells*, the only cells allowed to send objects to the "environment".

The essential novelty of our proposal is to replace the arbitrary arc set used in neural P systems by a more structured arc set δ (dag), or, otherwise interpreted, as a generalized multiset-based hypergraph. This interpretation has actually suggested the name of our proposal, hyperdag P systems, and their abbreviation hP systems.

The changes in the rules format are mostly adaptations needed by the new topological structure. Here, we have reused and enhanced the rewriting rules used by nP systems [11]. However, we could adopt and adapt any other rule set, from other variants or extensions of P systems, such as rewriting, symport/antiport or boundary rules [12].

Definitions of *configurations*, *transitions*, *computations* and *results of computations* in hP systems are similar to definitions used for nP systems (see also [8]), with the following essential additions and differences, here informally stated:

- The *rewriting mode* α and *transfer mode* β may not be fixed from the start, i.e., they may vary, for each cell σ_i and state $s \in Q_i$.
- If *object transfer mode* is *repl* (this is a deterministic step):
  - the objects tagged with $\uparrow$ will be sent to all the parents, replicated as necessary
  - the objects tagged with $\downarrow$ will be sent to all the children, replicated as necessary

  - the objects tagged with $\leftrightarrow$ will be sent to all the siblings, of all sibling groups, replicated as necessary

- If *object transfer mode* is *one* (this is a nondeterministic step):
  - the objects tagged with $\uparrow$ will be sent to one of the parents, arbitrarily chosen
  - the objects tagged with $\downarrow$ will be sent to one of the children, arbitrarily chosen
  - the objects tagged with $\leftrightarrow$ will be sent to one of the siblings, of one of the sibling groups, arbitrarily chosen

- If *object transfer mode* is *spread* (this is a nondeterministic step):
  - the objects tagged with $\uparrow$ will be split into submultisets and distributed among the parents, in an arbitrary way
  - the objects tagged with $\downarrow$ will be split into submultisets and distributed among the children, in an arbitrary way
  - the objects tagged with $\leftrightarrow$ will be split into submultisets and distributed among the siblings and sibling groups, in an arbitrary way

Figure 5 schematically shows the possible object transfers from a cell σ_i, having two children, two parents, hence two sibling groups, with one sibling in the first group and two siblings in the other. The above mentioned transfer modes will select one, some or all the illustrated transfer targets, deterministically (*repl*) or nondeterministically (*one, spread*).

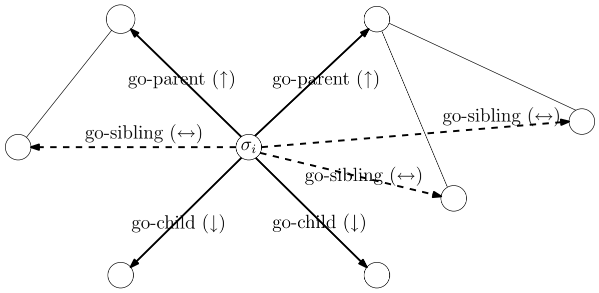


Figure 5: An annotated hP system indicating possible transfers from cell σ_i. The parent-child axis is top-down. Plain lines indicate parent-child relations and dashed lines indicate siblings. Arrows at the end of long thick lines, plain or dashed, indicate possible transfer directions from cell σ_i.

More formal definitions follow.

Definition 2 (Configurations) A *configuration* of the hP system Π is an m-tuple of the form $(s_1w_1,\dots,s_mw_m)$, with $s_i \in Q_i$ and $w_i \in O^*$, for $1 \le i \le m$. The m-tuple $(s_{1,0}w_{1,0},\dots,s_{m,0}w_{m,0})$ is the *initial configuration* of Π.

Definition 3 (Rewriting and transfer modes) For an hP system of degree m,

- the *object rewriting mode* is a function $\alpha : \bigcup_{i\in\{1,\dots,m\}} \{i\}\times Q_i \to \{min, par, max\}$.

- the *object transfer mode* is a function $\beta : \bigcup_{i\in\{1,\dots,m\}} \{i\}\times Q_i \to \{repl, one, spread\}$.

Definition 4 (Rewriting steps) For each cell σ_i with $s, s' \in Q_i$, $x \in O^*$, $y \in O^*_{tot}$, we define a *rewriting step*, denoted by $\Rightarrow_\alpha$, where $\alpha = \alpha(i,s) \in \{min, par, max\}$.

- $sx \Rightarrow_{min} s'y$ iff $sw \to s'w' \in P_i$, $w \subseteq x$, and $y = (x - w) \cup w'$;
- $sx \Rightarrow_{par} s'y$ iff $sw \to s'w' \in P_i$, $w^k \subseteq x$, $w^{k+1} \not\subseteq x$, for some $k \geq 1$, and $y = (x - w^k) \cup w'^k$;
- $sx \Rightarrow_{max} s'y$ iff $sw_1 \to s'w'_1, \ldots, sw_k \to s'w'_k \in P_i$, $k \geq 1$, such that $w_1 \ldots w_k \subseteq x, y = (x - w_1 \ldots w_k) \cup w'_1 \ldots w'_k$, and there is no $sw \to s'w' \in P_i$, such that $w_1 \ldots w_k w \subseteq x$ (note that rules can be combined only if they start from the same state s and end in the same state s').

Definition 5 (Transition steps) Given two configurations $C_1 = (s_1 w_1, \ldots, s_m w_m)$ and $C_2 = (s'_1 w''_1, \ldots, s'_m w''_m)$, we write $C_1 \Rightarrow_{\alpha,\beta} C_2$, for α and β (as defined in Definition 3), if the conditions below are met.

First, we apply rewriting steps (as defined in Definition 4) on each cell, i.e., $s_i w_i \Rightarrow_{\alpha(i,s_i)} s'_i w'_i$, $1 \leq i \leq m$.

Secondly, we define $z^{\uparrow}_{j,k}$, $z^{\downarrow}_{j,k}$, $z^{\leftrightarrow}_{j,k}$, the outgoing multisets from j to k, where $j \in \{1, \ldots, m\}$ and, respectively, $k \in \delta^{-1}(j)$, $k \in \delta(j)$, $k \in \delta(\delta^{-1}(j)) \setminus \{j\}$:

- If $\beta(j, s_j) = repl$, then
  - $z^{\uparrow}_{j,k} = w'_j|_{\uparrow}$, for $k \in \delta^{-1}(j)$;
  - $z^{\downarrow}_{j,k} = w'_j|_{\downarrow}$, for $k \in \delta(j)$;
  - $z^{\leftrightarrow}_{j,k} = w'_j|_{\leftrightarrow}$, for $k \in \delta(\delta^{-1}(j)) \setminus \{j\}$.
- If $\beta(j, s_j) = one$, then
  - $z^{\uparrow}_{j,k_j} = w'_j|_{\uparrow}$, for an arbitrary $k_j \in \delta^{-1}(j)$, and $z^{\uparrow}_{j,k} = \lambda$ for $k \in \delta^{-1}(j) \setminus \{k_j\}$;
  - $z^{\downarrow}_{j,k_j} = w'_j|_{\downarrow}$, for an arbitrary $k_j \in \delta(j)$, and $z^{\downarrow}_{j,k} = \lambda$ for $k \in \delta(j) \setminus \{k_j\}$;
  - $z^{\leftrightarrow}_{j,k_j} = w'_j|_{\leftrightarrow}$, for an arbitrary $k_j \in \delta(\delta^{-1}(j)) \setminus \{j\}$, and $z^{\leftrightarrow}_{j,k} = \lambda$ for $k \in \delta(\delta^{-1}(j)) \setminus \{j, k_j\}$.
- If $\beta(j, s_j) = spread$, then
  - $\{z^{\uparrow}_{j,k}\}_{k \in \delta^{-1}(j)}$ is an arbitrary multiset partition of $w'_j|_{\uparrow}$;
  - $\{z^{\downarrow}_{j,k}\}_{k \in \delta(j)}$ is an arbitrary multiset partition of $w'_j|_{\downarrow}$;
  - $\{z^{\leftrightarrow}_{j,k}\}_{k \in \delta(\delta^{-1}(j)) \setminus \{j\}}$ is an arbitrary multiset partition of $w'_j|_{\leftrightarrow}$.

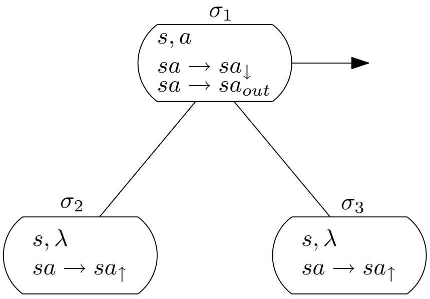
Finally, we set $w''_i = w'_i|_O \cup \bigcup_{j \in \delta^{-1}(i)} z^{\uparrow}_{j,i} \cup \bigcup_{j \in \delta(i)} z^{\downarrow}_{j,i} \cup \bigcup_{j \in \delta(\delta^{-1}(i)) \setminus \{i\}} z^{\leftrightarrow}_{j,i}$, for $i \in \{1, \ldots, m\}$.

Definition 6 (Halting and results) If no more transitions are possible, the hP system halts. For halted hP system, the *computational result* is the multiset that was cumulatively sent *out* (to the "environment") from the output cells I_{out}. The *numerical result* is the set of vectors consisting of the object multiplicities in the multiset result. Within the family of P systems, two systems are *functionally equivalent*, if they yield the same computational results.

Example 7 Consider two hP systems, Π_1 and Π_2, (which are functionally equivalent).

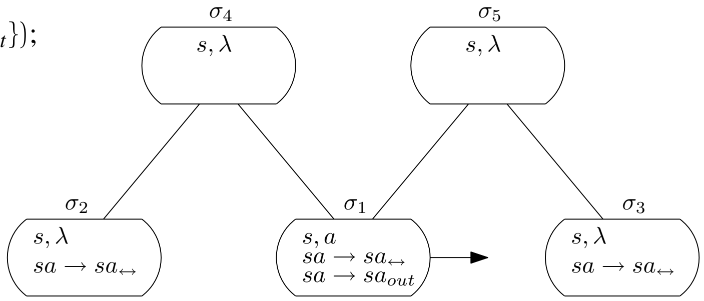
$\Pi_1 = (O, \sigma_1, \sigma_2, \sigma_3, \delta, I_{out})$, where:

- $O = \{a\}$;
- $\sigma_1 = (\{s\}, s, a, \{sa \to sa_{\downarrow}, sa \to sa_{out}\})$;
- $\sigma_2 = (\{s\}, s, \lambda, \{sa \to sa_{\uparrow}\})$;
- $\sigma_3 = (\{s\}, s, \lambda, \{sa \to sa_{\uparrow}\})$;
- $\delta = \{(1,2), (1,3)\}$;
- $I_{out} = \{1\}$.



$\Pi_2 = (O, \sigma_1, \sigma_2, \sigma_3, \sigma_4, \sigma_5, \delta, I_{out})$, where:

- $O = \{a\}$;
- $\sigma_1 = (\{s\}, s, a, \{sa \to sa_{\leftrightarrow}, sa \to sa_{out}\})$;
- $\sigma_2 = (\{s\}, s, \lambda, \{sa \to sa_{\leftrightarrow}\})$;
- $\sigma_3 = (\{s\}, s, \lambda, \{sa \to sa_{\leftrightarrow}\})$;
- $\sigma_4 = (\{s\}, s, \lambda, \emptyset)$;
- $\sigma_5 = (\{s\}, s, \lambda, \emptyset)$;
- $\delta = \{(4,1), (4,2), (5,1), (5,3)\}$;
- $I_{out} = \{1\}$.



4 Relations between P Systems, Neural P Systems and Hyperdag P Systems

Theorem 8 (Hyperdag P systems include non-dissolving transition P systems).
Any non-dissolving[1] transition P system can be simulated by an hP system, with the same number of steps and object transfers.

Proof: Given a non-dissolving, transition P system Π_T [12], we build a functionally equivalent hP system Π_H by the following transformation f. Essentially, we use the same elements, with minor adjustments. As the underlying structure, we can reuse the rooted tree structure of the P systems, because any rooted tree is a dag.
$\Pi_T = (O, C, \mu, w_1, \ldots, w_m, R_1, \ldots, R_m, i_o)$, $\Pi_H = f(\Pi_T) = (O', \sigma'_1, \ldots, \sigma'_m, \delta, I_{out})$.

- $O' = O$;
- $\sigma'_i = (Q'_i, s'_{i,0}, w'_{i,0}, P'_i)$, $1 \le i \le m$, where:
  - $Q'_i = \{s\}$, where $s \notin O$;
  - $s'_{i,0} = s$;
  - $w'_{i,0} = w_i$;
  - $P'_i = \{su \to sv' \mid u \to v \in R_i\}$, where v' is a translation of v by the following homomorphism: $(O \cup O \times \{here, in, out\})^* \to O^*$, such that $a \to a$, $(a, here) \to a$, $(a, out) \to a_{\uparrow}$, $(a, in) \to a_{\downarrow}$.

[1] A dissolving membrane occurs if we allow rules that tell a membrane to disappear where its remaining objects go to its parent membrane; see [12].

- $\delta = \mu$;
- $I_{out} = \{i_o\}$;
- The object rewriting mode is the *max* constant function, i.e., $\alpha(i,s) = max$, for $i \in \{1,\dots,m\}, s \in Q_i$;
- The object transfer mode is the *spread* constant function, i.e., $\beta(i,s) = spread$, for $i \in \{1,\dots,m\}, s \in Q_i$.

Tags go-child(↓), go-parent(↑) correspond to P system target indications *in*, *out*, respectively. An empty tag corresponds to P system target indication *here*. Object rewriting and transfer modes of hP systems are a superset of object rewriting and transfer modes of P systems.

We omit here the rest of the proof that shows the two systems, Π_T and Π_H, yield the same computational results, which is now straightforward but lengthy. □

Remark 9 In analogy to the P systems, it is straightforward to extend the hP systems with additional features, such as dissolving membranes, priorities or polarities. However, to keep the arguments simple, we here omit such extensions.

Proving that hP systems also simulate nP systems appears more daunting. However, here we will use a natural interpretation of hP systems, where the bulk of the computing will be done by the sink nodes, and the upper nodes (parents) will function mostly as communication channels.

Remark 10 The combination of *go-sibling* (↔) with *repl* object transfer mode enables the efficient modelling of a communication *bus*, using only one hyperedge or, in the corresponding dag, *n* arcs. In contrast, any formal systems that use graph edges (or digraph arcs) to model 1:1 communication channels will need $n(n-1)$ separate edges (or $2n(n-1)$ arcs) to model the associated complete subgraph (clique). It is expected that this modelling improvement will also translate into a complexity advantage, if we count the number of messages. In hP systems, a local broadcast needs only one message to siblings, while graph- or digraph-based systems need $n-1$ messages.

Example 11 Figure 6 shows the structure of an hP system that models a computer network. Four computers are connected to "Ethernet Bus 1", the other four computers are connected to "Ethernet Bus 2", while two of the first group and two of the second group are at the same time connected to the same wireless cell. In this figure, we also suggest that "Ethernet Bus 1" and "Ethernet Bus 2" are themselves connected to a higher level communication hub in a generalized hypergraph.

Example 12 Figure 7 shows the computer network of Figure 6, modelled as a graph (if we omit arrows) or as a digraph (if we consider the arrows). Note that the graph- or digraph-based models, such as nP systems, do not support the grouping concept, i.e., there is no direct way to mark the nodes a,b,c,d as being part of the "Ethernet Bus 1".

We can now mention an important theorem comparing hP systems and nP systems.

Theorem 13 (Hyperdag P systems can simulate symmetric neural P systems).
Any symmetric nP system can be simulated by an hP system, with the same number of steps and object transfers.

Proof: The simulation details are given in our research report [8]. □

Remark 14 We leave here open the case of non-symmetric nP systems, which can also be simulated by hP systems, but require additional costs, in terms of steps and object transfers.

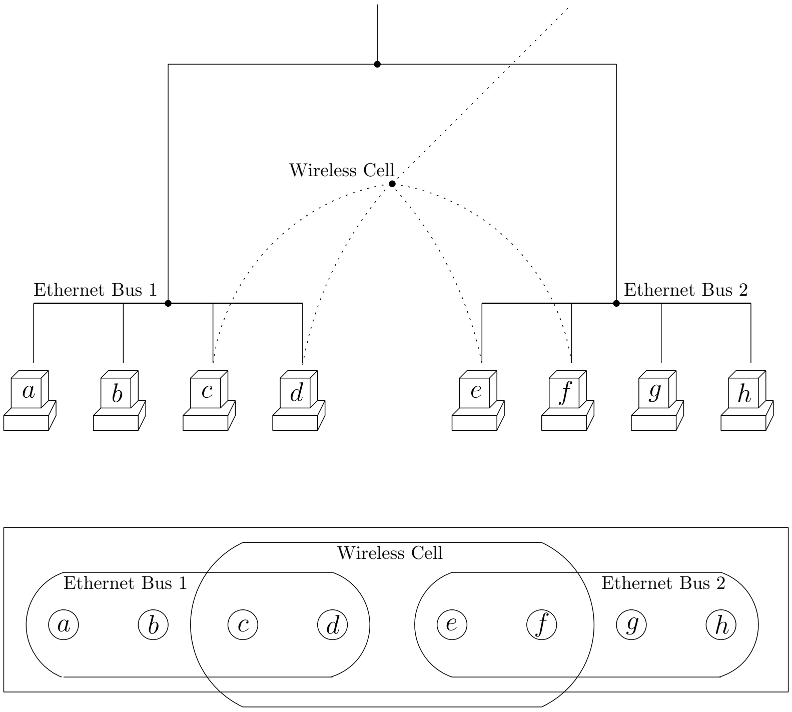


Figure 6: A computer network and its corresponding hypergraph representation.

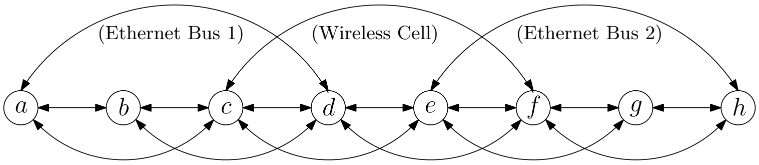


Figure 7: The digraph representation of the computer network of Figure 6.

5 Planar representation of hyperdag P Systems

Classical tree-based P systems allow a "nice" planar representation, where the parent–child relationships between membranes are represented by Venn-like diagrams. Can we extend this representation to cover our dag-based hP systems?

In this section, we will show that any hP system, structurally based on a canonical dag, can still be *intensionally* represented by hierarchically nested planar regions, delimited by Jordan curves (simple closed curves). Conversely, we also show that any set of hierarchically nested planar regions delimited by Jordan curves can be interpreted as a canonical dag, which can form the structural basis of a number of hP systems.

We will first show how to represent a canonical dag as a set of hierarchically nested planar regions.

Algorithm 15 (Algorithm for visually representing a canonical dag)

Without loss of generality, we consider a canonical dag (V,δ) of order n, where vertices are indexed according to an arbitrary topological order implied by the arcs, by considering parents before the children, i.e., $V = \{v_i \mid 1 \leq i \leq n\}$, where $(v_i, v_j) \in \delta$ implies $i < j$. Figure 8 shows side by side a simple height 1 canonical dag and its corresponding hypergraph representation. Note the *intensional* representation (as opposed to the *extensional* one), where v_2 is not totally included in v_1, although all vertices included in v_2, i.e., v_4 and v_5, are also included in v_1. A possible topological order is v_1, v_2, v_3, v_4, v_5.

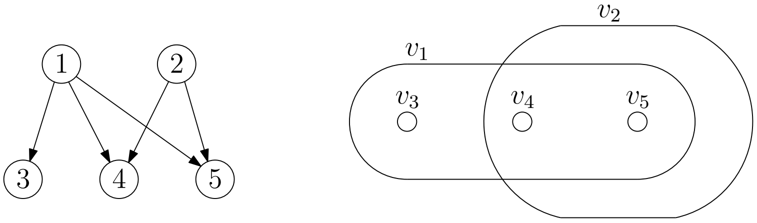


Figure 8: A simple canonical dag and its corresponding hypergraph representation.

For each vertex v_i, we associate a distance $\psi_i = \frac{1}{2^{(n-i+1)}}$, for $i \in \{1,\ldots,n\}$. For Figure 8, $\psi_i = \frac{1}{32}, \frac{1}{16}, \frac{1}{8}, \frac{1}{4}, \frac{1}{2}$, for $i \in \{1,\ldots,n\}$.

We process the vertices in reverse topological order $v_n,\ldots,v_1$, at each step i representing the current vertex v_i by a planar region R_i.

First, we set parallel horizontal axis X_o and X_p, vertically separated by distance $3(n-1)$. Secondly, we set points $o_1,\ldots,o_n$ on X_o, such that o_i and o_{i+1} are separated by distance 3, for $1 \le i \le n-1$. We define o_i as the *origin point* of v_i, and write $o_i = origin(v_i)$. Finally, we set points $p_1,\ldots,p_n$ on X_p, such that p_i and p_{i+1} are separated by distance 3, for $1 \le i \le n-1$. We define p_i as the *corridor point* of v_i.

Figure 9 shows the construction of X_o, X_p, o_i and p_i, for the dag of Figure 8, where $n = 5$.

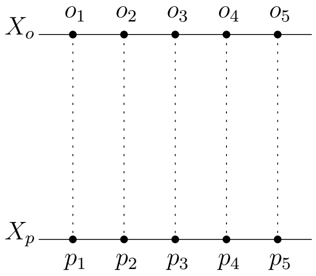


Figure 9: Construction of X_o, X_p, o_i and p_i, for the dag of Figure 8, where $n = 5$.

If the current vertex v_i is a sink, then R_i is a circle with radius $\frac{1}{2}$ centered at o_i.

If the current vertex v_i is a non-sink, then R_i is constructed as follows. Assume that the children of v_i are $w_1,\ldots,w_{n_i}$, and their (already created) regions are $S_1,\ldots,S_{n_i}$. Consider line segments $l_0, l_1,\ldots,l_{n_i}$, where l_0 is bounded by o_i and p_i, and l_j is bounded by p_i and $origin(w_j)$, for $j \in \{1,\ldots,n_i\}$. Let L_0, $L_1,\ldots,L_{n_i}$, $T_1,\ldots,T_{n_i}$ be the regions enclosed by Jordan curves around l_0, $l_1,\ldots,l_{n_i}$, $S_1,\ldots,S_{n_i}$, at a distance ψ_i, and let $R_i' = L_0 \cup \bigcup_{j=1,\ldots,n_i} L_j \cup \bigcup_{j=1,\ldots,n_i} T_j$. We define R_i as the external contour of R_i'. This definition will discard all internal holes (such as the dashed enclosed regions of Figure 10), if any, without introducing any additional containment relations between our regions. The details of our construction guarantee that no internal hole will ever contain an origin point. □

Figure 10 shows the side by side, a dag and its corresponding planar region representation; internal holes are represented by dotted lines. Our objective here was not to create "nice" visualizations, but to prove that it is possible to represent an arbitrary canonical dag, i.e., an arbitrary hP system structurally based on a canonical dag, by hierarchically nested planar regions.

We will next show that, for any finite set of hierarchically nested planar regions, we can build a corresponding canonical dag (i.e., the underlying structure of an hP system).

Algorithm 16 (Algorithm for building a canonical dag from finite set of hierarchically nested planar regions)

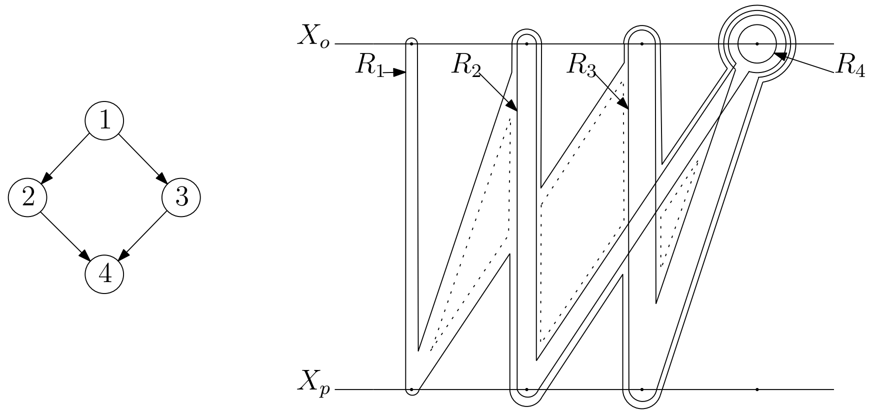


Figure 10: A height 2 dag and its corresponding representation, built by Algorithm 15.

Assume that we have n hierarchically nested planar regions,

1. Label each planar region by R_i, $i \in \{1,\ldots,n\}$,
2. If R_i directly nests R_j then draw an arc from a vertex v_i to a vertex v_j, $i, j \in \{1,\ldots,n\}$, $i \neq j$.

□

We now show that a canonical digraph produced from Algorithm 16 does not contain any cycles. Our proof is by contradiction. Let us assume a digraph G produced from Algorithm 16 contains a cycle $v_i,\ldots,v_k,\ldots,v_i$. Then every vertex in a cycle has an incoming arc. If vertex v_k is a maximal element in a cycle, with respect to direct nesting, then its corresponding planar region R_k have the largest region area among planar regions in a cycle. Since no other planar region in a cycle can contain R_k, there are no arc incident to vertex v_k. Hence, there is no cycle in G.

Remark 17 We present in [9] a solution to the problem of representing dags (that contain transitive arcs) by a set of simple regions, where direct containment denotes a parent–child relation.

6 Summary

We have proposed a new model, as an extension of P systems, that provides a better communication structure and we believe is often more convenient for modelling real-world applications based on tree structures augmented with secondary or shared communication channels.

We have shown that hP systems functionally extends the basic functionality of transition P systems and nP systems, even though the underlying structure of hP systems is different. In dag-based hP systems, we can have a natural separation of computing cells (sinks) from communication cells (hyperedges). This model also allows us to easily represent multiple inheritance or to distribute computational results (as specified by a dag) amongst several different parts of a membrane structure.

We note that the operational behavior of hP systems is separate from the topological structure of a membrane system. In this paper, we illustrated hP systems using the computational rules of nP systems, where multisets of objects are repeatedly changed within cells, by using a fixed set of multiset rewriting rules, or transferred between cells, using several possible transfer modes.

Finally, we provided an intuitive visualization of hP systems, by showing that any set of hierarchically nested planar regions (which represents any set of cells ordered by containment) is equivalent to, or modelled by, a dag without transitive arcs. We provided simple algorithms to translate between these two interpretations.

This paper is part of an ongoing research dedicated to structured modelling and model checking of P systems.

Dedication

This article is dedicated to Mario J. Pérez-Jiménez, on the occasion of his 60th birthday (November 2008).

Bibliography

[1] C. Berge, *Hypergraphs: Combinatorics of Finite Sets*, Elsevier Science Publishers, 1989.

[2] C. Carathéodory, *Theory of Functions of a Complex Variable*, Vol.1, Chelsea Publishing Company, 1954.

[3] G. Ciobanu, Distributed Algorithms Over Communicating Membrane Systems, *Bio Systems*, 70(2):123–133, 2003.

[4] W. F. Doolittle, Uprooting the Tree of Life, *Scientific American*, 282(2):90–95, 2000.

[5] T. -O. Ishdorj, M. Ionescu, Replicative-Distribution Rules in P Systems with Active Membranes, *Proceeding of the First International Colloquium on Theoretical Aspects of Computing (ICTAC 2004)*, 68–83, 2004.

[6] C. Li, *Validating P System as Distributed Computing Models*, Master thesis, 2008.

[7] C. Martín-Vide, G. Păun, J. Pazos, A. Rodríguez-Patón, Tissue P Systems, *Theoretical Computer Science*, 296(2):295–326, 2003.

[8] R. Nicolescu, M. J. Dinneen, Y.-B. Kim, Structured Modelling with Hyperdag P Systems: Part A, CDMTCS Research Report Series, *CDMTCS-342*, 1–24, December 2008. `http://www.cs.auckland.ac.nz/CDMTCS/researchreports/342hyperdagA.pdf`

[9] R. Nicolescu, M. J. Dinneen, Y.-B. Kim, Discovering the Membrane Topology of Hyperdag P Systems, *Proceeding of the Tenth Workshop on Membrane Computing (WMC10 2009)*, Curtea de Argeş, Romania, August 24–27, 426–451, 2009.

[10] G. Păun, Computing with Membranes. *Journal of Computer and System Sciences*, 61(1):108–143, 2000, (and Turku Center for Computer Science, TUCS Report 208, November 1998).

[11] G. Păun, *Membrane Computing: An Introduction*. Springer-Verlag, 2002.

[12] G. Păun, Introduction to Membrane Computing, *Proceeding of the First Brainstorming Workshop on Uncertainty in Membrane Computing*, 17–65, 2004.

[13] G. Păun, R. A. Păun, Membrane Computing as a Framework for Modeling Economic Processes, *Seventh International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC 2005)*, 11–18, 2005.

[14] M. Slikker, A. Van Den Nouweland, *Social and Economic Networks in Cooperative Game Theory*, Kluwer Academic Publishers, 2001.

[15] V. I. Voloshin, *Coloring Mixed Hypergraphs: Theory, Algorithms and Applications*, American Mathematical Society, 2002.

[16] C. Zandron, C. Ferretti, G. Mauri, Solving NP-Complete Problems Using P Systems with Active Membranes, *Proceeding of the Second International Conference on Unconventional Models of Computation*, 289–301, 2000.

Michael J. Dinneen received his PhD in 1996 from the University of Victoria in Canada. He is currently a Senior Lecturer at the University of Auckland. His research interests are in combinatorial algorithms, graph algorithms, uncoventional computing models and network design.

Yun-Bum Kim is currently a PhD student at the University of Auckland, New Zealand. In 2007 he has completed a MSc thesis in network optimization design problems and currently has interests in molecular and distributed computing.

Radu Nicolescu (PhD Bucharest, MACM, MemIEEE) is currently a Senior Lecturer at the University of Auckland, New Zealand. He has research interests in formal languages, information complexity and service oriented computing.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 2, pp. 238-250

Solving Problems in a Distributed Way in Membrane Computing: dP Systems

Gh. Păun, M.J. Pérez-Jiménez

Gheorghe Păun
Institute of Mathematics of the Romanian Academy
PO Box 1-764, 014700 Bucureşti, Romania, and

Department of Computer Science and Artificial Intelligence
University of Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
E-mail: gpaun@us.es

Mario J. Pérez-Jiménez
Department of Computer Science and Artificial Intelligence
University of Sevilla
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain
E-mail: marper@us.es

Abstract: Although P systems are distributed parallel computing devices, no explicit way of handling the input in a distributed way in this framework was considered so far. This note proposes a distributed architecture (based on cell-like P systems, with their skin membranes communicating through channels as in tissue-like P systems, according to specified rules of the antiport type), where parts of a problem can be introduced as inputs in various components and then processed in parallel. The respective devices are called dP systems, with the case of accepting strings called dP automata. The communication complexity can be evaluated in various ways: *statically* (counting the communication rules in a dP system which solves a given problem), or *dynamically* (counting the number of communication steps, of communication rules used in a computation, or the number of objects communicated). For each measure, two notions of "parallelizability" can be introduced. Besides (informal) definitions, some illustrations of these idea are provided for dP automata: each regular language is "weakly parallelizable" (i.e., it can be recognized in this framework, using a constant number of communication steps), and there are languages of various types with respect to Chomsky hierarchy which are "efficiently parallelizable" (they are parallelizable and, moreover, are accepted in a faster way by a dP automaton than by a single P automaton). Several suggestions for further research are made.

Keywords: Membrane computing, P system, distributed computing, communication complexity, Chomsky hierarchy.

1 Introduction

P systems are by definition distributed parallel computing devices, [11], [12], [17], and they can solve computationally hard problems in a feasible time, [13], but this efficiency is achieved by a trade-off between space and time, based on the possibility of generating an exponential workspace in a linear time, by means of biologically inspired operations, such as membrane division and membrane creation. However, no class of P systems was proposed where a hard problem can be solved in a distributed parallel way after splitting the problem in parts and introducing these subproblems in components of a P system which can work on these subproblems in parallel and produce the solution to the initial problem by interacting/communicating among each other (like in standard distributed computer science). In particular, no communication complexity, in the sense of [2], [9], [16], was considered for P systems, in spite of the fact that computation (time) complexity is very well developed, [13], and also space complexity was recently investigated, [14]. Some proposals towards a communication complexity of P systems were made in [1], but mainly related to the communication effort in terms of symport/antiport rules used in so-called evolution-communication P systems of [5]. (Note that in communication complexity theory the focus is not on the time efficiency of solving a problem, but the parties involved in the computation just receive portions of the input, in general, distributed in a balanced manner, “as fair as possible" – this distribution introduces an inherent difficulty in handling the input – and then mainly the complexity of the communication needed to parties to handle this input is investigated.)

This note tries to fill in this gap, proposing a rather natural framework for solving problems in a distributed way, using a class of P systems which mixes ingredients already existing in various much investigated types of P systems. Namely, we consider P systems with inputs, in two variants: (i) like in P automata, [6], [10], where a string of symbols is recognized if those symbols are brought into the system from the environment and the computation eventually halts (it is important to note that the string is “read" during the computation, not *before* it), and (ii) in the usual manner of complexity investigations, [13], where an instance of a decision problem is introduced in a P system in the form of a multiset of symbols (this operation takes no time, the computation starts *after* having the code of the problem inside), and the system decides that instance in the end of a computation which sends to the environment one of the special objects `yes` or `no`. Several such systems, no matter of what type, are put together in a complex system which we call *dP system* (from “distributed P system"); the component systems communicate through channels linking their skin membranes, by antiport rules as in tissue-like P systems. When accepting strings by dP systems with P automata as components, the device is called a *dP automaton*.

Such an architecture was already used, with specific ingredients, for instance, in the investigations related to eco-systems, where “local environments" are necessary to be delimited and communication possibilities exist, linking them; details can be found in the recent paper [4].

The way to use a dP system is obvious: a problem Q is split into parts $q_1, q_2, \ldots, q_n$, which are introduced in the n components of the dP system (as in P automata or as in decision P systems), these n systems work separately on their problems, and communicate to each other according to the skin-to-skin rules. The solution to the problem Q is provided by the whole system (by halting – in the case of accepting strings, by sending out one of the objects `yes` or `no`, etc.). Like in communication complexity, [9], we request the problem to be distributed in a *balanced* way among the components of the dP system, i.e., in “as equal as possible" parts (also an *almost balanced* way to distribute the input among two processors is considered in [9] – no partner takes more than two thirds of the input – which does not seem very natural to be extended to the general case, of n processors).

Several possibilities exist for defining the communication complexity of a computation. We follow here the ideas of [1], and introduce three measures: the number of steps of the computation when a communication rule is used (such a step is called communication step), the number of communication rules used during a computation, and the number of objects transferred among components (by communication rules) during a computation. All these three measures are dynamically defined; we can also consider a static parameter, like in descriptional complexity of Chomsky languages (see a survey in [8]), i.e., the number of communication rules in a dP system.

A problem is said to be “weakly parallelizable" with respect to a given (dynamical) communication complexity measure if it can be split in a balanced way, introduced in the dP system, and solved using a number of communication steps bounded by a constant given in advance; a problem is “efficiently parallelizable" if it is weakly parallelizable and can be solved by a dP system in a more efficient way than by a single P system; more precise definitions are given in the next sections of the paper.

Various possibilities exist, depending on the type of systems (communicating systems, e.g., based on symport/antiport rules, systems with active membranes, catalytic systems, etc.) and the type of problem we consider (accepting strings, decision problems, numerical problems, etc.).

In this note we only sketch the general formal framework and give an illustration, for the case of accepting strings as in P automata. We only show here that all regular languages are weakly parallelizable (only one communication step suffices, hence the weak parallizability holds with respect to all three dynamical measures), and that there are regular, context-free non-regular, context-sensitive non-context-free languages which are efficiently parallelizable with respect to the first two dynamical measures mentioned above (in view of the results in [9], there are linear languages which are not efficiently parallelizable with respect to the number of communicated objects/bits among components).

If we use extended systems (a terminal alphabet of objects is available) and the communication channels among the components of a dP automaton are controlled, e.g., by states, as in [7], or created during the computation, as in [3], then the power of our devices increases considerably: all recursively enumerable languages are weakly parallelizable in this framework.

Many research problems remain to be explored, starting with precise definitions for given classes of P systems, continuing with the study of usefulness of this strategy for solving computationally hard problems (which problems are weakly/efficiently parallelizable and which is the obtained speed-up for them?), and ending with a communication complexity theory of dP systems, taking into account all measures of complexity mentioned above (for the number of objects communicated among components, which corresponds to the number of bits considered in [9], we can transfer here the general results from communication complexity – note however that in many papers in this area one deals with 2-party protocols, while in our framework we want to have an n-party set-up, and that we are also interested in the time efficiency of the distributed and parallel way of solving a problem).

2 dP Systems – A Preliminary Formalization

The reader is assumed familiar with basics of membrane computing, e.g., from [11], [12], and of formal language theory, e.g., from [15], hence we pass directly to introducing our proposal of a distributed P system. The general idea is captured in the following notion.

A *dP scheme* (of degree $n \geq 1$) is a construct

$$\Delta = (O, \Pi_1, \ldots, \Pi_n, R),$$

where:

1. O is an alphabet of objects;
2. $\Pi_1, \ldots, \Pi_n$ are cell-like P systems with O as the alphabet of objects and the skin membranes labeled with $s_1, \ldots, s_n$, respectively;
3. R is a finite set of rules of the form $(s_i, u/v, s_j)$, where $1 \leq i, j \leq n$, $i \neq j$, and $u, v \in O^*$, with $uv \neq \lambda$; $|uv|$ is called the *weight* of the rule $(s_i, u/v, s_j)$.

The systems $\Pi_1, \ldots, \Pi_n$ are called *components* of the scheme Δ and the rules in R are called *inter-components communication rules*. Each component can take an input, work on it, communicate with other components (by means of rules in R), and provide the answer to the problem in the end of a halting computation. (A delicate issue can appear in the case of components which can send objects to the environment and bring objects from the environment – this happens, for instance, for symport/antiport P systems; in this case we have to decide whether or not the components can exchange objects by means of the environment, or the only permitted communication is done by means of the rules in R. For instance, a "local environment" for each component can be considered, disjoint from the "local environments" of other components, thus preventing the interaction of components by means of other rules than those in R. Actually, the rules in R themselves can be defined between these "local environments" – which is a variant worth to explore. We point out here that also the need of a "local environment" has appeared in the applications of membrane computing to eco-systems investigations, see [4] and its references.)

Now, we can particularize this notion in various ways, depending on the type of systems $\Pi_i, 1 \leq i \leq n$, and the type of problems we want to solve.

For instance, we can define *dP systems with active membranes*, as dP schemes as above, with the components being P systems with active membranes, each of them having a membrane designated as the input membrane. Having a decision problem – consider, e.g., SAT for n variables and m clauses – we can split a given instance of it in parts which are encoded in multisets which are introduced in the components of the dP system. For example, we can introduce the code of each separate clause in a separate component of the dP system. The components start to work, each one deciding its clause, and in the end they communicate to each other the result; if one of the components will find that all m clauses are satisfied, then the whole SAT formula is satisfied. Intuitively, this is a faster way than deciding the formula by means of a single P system with active membranes – but a crucial aspect has been neglected above: in order to say that the formula is satisfied, all the m clauses should be satisfied *by the same truth-assignment*, and this supposes that the m components communicate to each other also which is the assignment which turns true the clauses. That is, besides the usual time complexity of solving the problem we have now to consider the cost of communication among the components and the trade-off between these two parameters should be estimated.

Another interesting case, which will be briefly investigated in the subsequent section, is that of accepting strings in the sense of P automata, [6], [10]; we will come back immediately to this case.

On the other hand, we have several possibilities for estimating "the cost of communication", and we adapt here the ideas from [1].

Let us consider a dP system Δ, and let $\delta : w_0 \Longrightarrow w_1 \Longrightarrow \ldots \Longrightarrow w_h$ be a halting computation in Δ, with w_0 being the initial configuration. Then, for each $i = 0, 1, \ldots, h-1$ we can write:

$ComN(w_i \Longrightarrow w_{i+1}) = 1$ if a communication rule is used in this transition, and 0 otherwise,

$ComR(w_i \Longrightarrow w_{i+1}) =$ the number of communication rules used in this transition,

$ComW(w_i \Longrightarrow w_{i+1}) =$ the total weight of the communication rules used in this transition.

These parameters can then be extended in the natural way to computations, results of computations, systems, problems/languages. We consider below the case of accepting strings (by $L(\Delta)$ we denote the language of strings accepted by Δ): for $ComX \in \{ComN, ComR, ComW\}$ we define

$ComX(\delta) = \sum_{i=0}^{h-1} ComX(w_i \Longrightarrow w_{i+1})$, for $\delta : w_0 \Longrightarrow w_1 \Longrightarrow \ldots \Longrightarrow w_h$ a halting computation,

$ComX(w, \Delta) = \min\{ComX(\delta) \mid \delta : w_0 \Longrightarrow w_1 \Longrightarrow \ldots \Longrightarrow w_h$ is a computation in Δ which accepts the string $w\}$,

$ComX(\Delta) = \max\{ComX(w, \Delta) \mid w \in L(\Delta)\}$,

$ComX(L) = \min\{ComX(\Delta) \mid L = L(\Delta)\}$.

Similar definitions can be considered for more general decidability problem than accepting strings, then complexity classes can be defined. We do not enter here into details for this general case; in the next section we will briefly consider the specific case of dP automata and of languages.

The previously sketched approach should be investigated in more details. Which is the speed-up for a given problem or class of problems? Clearly, $ComN(\alpha) \leq ComR(\alpha) \leq ComW(\alpha)$, for all valid

α. Moreover, in one communication step one can use arbitrarily many communication rules, which therefore move from a component to another one arbitrarily many objects. Anyway, independently of the communication cost, presumably, only a linear speed-up can be obtained by splitting the problem in a given number of parts. Are there problems which however cannot be solved in this framework in a faster way than by using a single P system (with active membranes) provided that the communication cost is bounded (e.g., using communication rules in R only for a constant number of times)? Which is the communication complexity for a given problem or class of problems? Finding suggestive examples can be a first step in approaching such issues.

A case study will be considered in the next section, not for dP systems with active membranes (which, we believe, deserve a separate and detailed examination), but for a distributed version of P automata.

3 dP Automata

We consider now the distributed version of P automata, [6], [10], which are symport/antiport P systems which accept strings: the sequence of objects (because we work with strings and symbol objects, we use interchangeably the terms “object" and “symbol") imported by the system from the environment during a halting computation is the string accepted by that computation (if several objects are brought in the system at the same time, then any permutation of them is considered as a substring of the accepted string; a variant, considered in [6], is to associate a symbol to each multiset and to build a string by such “marks" attached to the imported multisets). The accepted string can be introduced in the system symbol by symbol, in the first steps of the computation (if the string is of length k, then it is introduced in the system in the first k steps of the computation – the P automaton is then called *initial*), or in arbitrary steps. Of course, the initial mode is more restrictive – but we do not enter here into details.

As a kind of mixture of the ideas in [6] and [10] for defining the accepted language, we can consider extended P automata, that is, with a distinguished alphabet of objects, T, whose elements are taken into account when building the accepted string (the other objects taken by the system from the environment are ignored). Here, however, we work with non-extended P automata.

A *dP automaton* is a construct

$$\Delta = (O, E, \Pi_1, \ldots, \Pi_n, R),$$

where $(O, \Pi_1, \ldots, \Pi_n, R)$ is a dP scheme, $E \subseteq O$ (the objects available in arbitrarily many copies in the environment), $\Pi_i = (O, \mu_i, w_{i,1}, \ldots, w_{i,k_i}, E, R_{i,1}, \ldots, R_{i,k_i})$ is a symport/antiport P system of degree k_i (without an output membrane), with the skin membrane labeled with $(i,1) = s_i$, for all $i = 1, 2, \ldots, n$.

A halting computation with respect to Δ accepts the string $x = x_1 x_2 \ldots x_n$ over O if the components $\Pi_1, \ldots, \Pi_n$, starting from their initial configurations, using the symport/antiport rules as well as the inter-components communication rules, in the non-deterministically maximally parallel way, bring from the environment the substrings $x_1, \ldots, x_n$, respectively, and eventually halts.

The dP automaton is synchronized, a universal clock exists for all components, marking the time in the same way for the whole dP automaton. It is also important to note that we work here in the non-extended case, all input symbols are recorded in the string. In this way, at most context-sensitive languages can be recognized.

The three complexity measures $ComN, ComR, ComW$ defined in the previous section can be directly introduced for dP automata (and they were formulated above for this case). With respect to them, we can consider two levels of parallelizability.

A language $L \subseteq V^*$ is said to be (n,m)-*weakly ComX parallelizable*, for some $n \geq 2, m \geq 1$, and $X \in \{N, R, W\}$, if there is a dP automaton Δ with n components and there is a finite subset F_Δ of L such that each string $x \in L - F_\Delta$ can be written as $x = x_1 x_2 \ldots x_n$, with $||x_i| - |x_j|| \leq 1$ for all $1 \leq i, j \leq n$, each component Π_i of Δ takes as input the string $x_i, 1 \leq i \leq n$, and the string x is accepted by Δ by a halting computation δ such that $ComX(\delta) \leq m$. A language L is said to be *weakly ComX parallelizable* if it is (n,m)-weakly *ComX* parallelizable for some $n \geq 2, m \geq 1$.

Two conditions are here important: (i) the string is distributed in equal parts, modulo one symbol, to the components of the dP automaton, and (ii) the communication complexity, in the sense of measure *ComX*, is bounded by the constant m.

We have said nothing before about the length of the computation. That is why we also introduce a stronger version of parallelizability.

A language $L \subseteq V^*$ is said to be (n,m,k)*-efficiently ComX parallelizable*, for some $n \geq 2, m \geq 1, k \geq 2$, and $X \in \{N,R,W\}$, if it is (n,m) weakly ComX parallelizable, and there is a dP automaton Δ such that

$$\lim_{x \in L, |x| \to \infty} \frac{time_{\Pi}(x)}{time_{\Delta}(x)} \geq k,$$

for all P automata Π such that $L = L(\Pi)$ ($time_{\Gamma}(x)$ denotes here the smallest number of steps needed for the device Γ to accept the string x). A language L is said to be *efficiently ComX parallelizable* if it is (n,m,k)-efficiently *ComX* parallelizable for some $n \geq 2, m \geq 1$, $k \geq 2$.

Note that in the case of dP automata, the duration of a computation may also depend on the way the string is split in substrings and introduced in the components of the system; in a natural way, one of the most efficient distribution of the string and shortest computation are chosen. Of course, as larger the constant k as better.

Moreover, while $time_{\Delta}(x)$ is just given by means of a construction of a suitable dP automaton Δ, $time_{\Pi}(x)$ should be estimated *with respect to all P automata* Π.

An example is worth considering in order to illustrate this definition. Let us examine the dP system from Figure 1 – the alphabet of objects is $O = \{a,b,c,d,c_1,c_2,\#\}$, and $E = \{a,b\}$.

Clearly, component Π_1 (in the left hand side of the figure) can only bring objects a,c inside, and component Π_2 (in the right hand side of the figure) can only bring objects b,d inside. In each step, only one of a,c, alternately, enters Π_1 and only one of b,d, alternately, enters Π_2 (note that we do not need objects c,d to be present initially in the environment, while one copy of each a and b is sufficient). The computation of each component can stop only by "hiding" the "carrier objects" c,d inside an inner membrane, and this means releasing c_1 in Π_1 and c_2 in Π_2. If these objects are not released at the same time in the two components, so that the exchange rule $(s_1, c_1/c_2, s_2)$ can be used, then, because of the maximal parallelism, the object c_1 should enter membrane (1,3), and object c_2 should enter membrane (2,3); in each case, the trap-object # is released, and the computation never stops: the object # oscillates forever across membrane (1,2) in Π_1 and across membrane (2,2) in Π_2.

Consequently, the two strings accepted by the two components of Δ should have the same length, that is the language accepted by the system is

$$L(\Delta) = \{(ac)^s (bd)^s \mid s \geq 0\}.$$

Note the crucial role played here by the fact that the system is synchronized, and that a computation which accepts a string $x_s = (ac)^s(bd)^s$, hence of length $4s$, lasts $2s+2$ steps ($2s$ steps for bringing objects inside, one step when objects c,d are introduced in an inner membrane, and one inter-components communication step), with one of these steps being a communication between components.

Obviously, if we recognize a string $x_s = (ac)^s(bd)^s$ as above by means of a usual symport/antiport P system, then, because no two symbols of the string can be interchanged, no two adjacent symbols can be introduced in the system at the same step, hence the computation lasts at least as many steps as the length of the string, that is, $4s$. This shows that our language is not only $(2,r)$-weakly *ComX* parallelizable, but also $(2,r,2)$-efficiently *ComX* parallelizable, for $(r,X) \in \{(1,N),(1,R),(2,W)\}$.

This conclusion is worth formulating as a theorem.

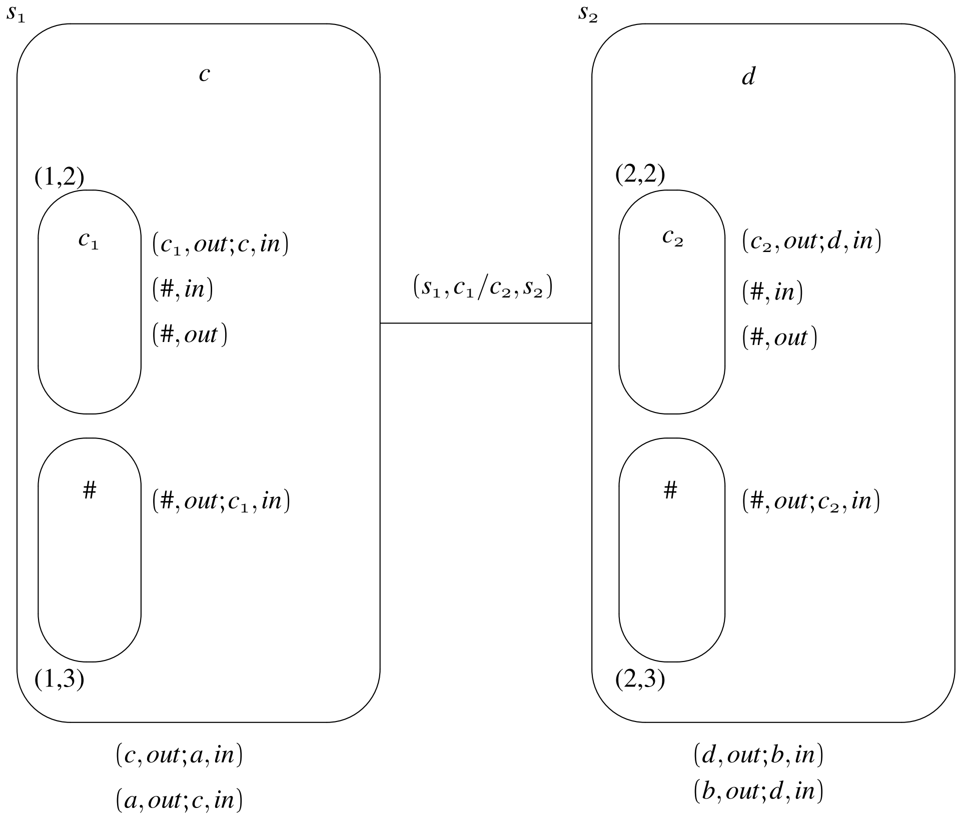


Figure 1: An example of a dP automaton

Theorem 1. *The language $L = \{(ac)^s(bd)^s \mid s \geq 0\}$ is efficiently ComX parallelizable, for all $X \in \{N,R,W\}$.*

Note that this language is not regular (but it is linear, hence also context-free).

The previous construction can be extended to dP automata with three components: Π_1 inputs the string $(ac)^s$, Π_2 inputs $(bd)^s$, and Π_3 inputs $(ac)^s$, then Π_1 produces the object c_1, Π_2 produces two copies of c_2, and Π_3 produces the object c_3. Now, c_1 is exchanged for one copy of c_2 from Π_2 and c_3 for the other copy, otherwise the computation never stops. The recognized language is $\{(ac)^s(bd)^s(ac)^s \mid s \geq 0\}$.

This language is not context-free, hence we have:

Theorem 2. *There are context-sensitive non-context-free languages which are efficiently ComX parallelizable, for all $X \in \{N,R,W\}$.*

The previous two theorems show that the distribution, in the form of dP systems, is useful from the time complexity point of view, although only one communication step is performed and only one communication rule is used at that step. Moreover, the proofs of the two theorems show that, in general, languages consisting of strings with two well related halves (but not containing “too much" information in each half of the string, besides the length), are weakly parallelizable, and, if no two adjacent symbols of the strings can be interchanged, then these languages are efficiently parallelizable.

We have said nothing above about regular languages – this is the subject of the next section.

4 All Regular Languages are Weakly Parallelizable

The assertion in the title of this section corresponds to Theorem 2.3.5.1 in [9], which states that for each regular language there is a constant k which bounds its (2-party) communication complexity. The version of this result in terms of weak *ComX* parallelizability is shown by the following construction. Consider a non-deterministic finite automaton $A = (Q,T,q_0,F,P)$ (set of states, alphabet, initial state, final states, set of transition rules, written in the form $qa \rightarrow q'$, for $q,q' \in Q, a \in T$). Without any loss of generality, we may assume that all states of Q are reachable from the initial state (for each $q \in Q$ there is $x \in T^*$ such that $q_0x \Longrightarrow^* q$ with respect to transition rules in P). We construct the following dP automaton:

$$\begin{array}{rcl}
\Delta & = & (O,E,\Pi_1,\Pi_2,R), \text{ where :}\\
O & = & Q \cup T \cup \{d\}\\
 & \cup & \{(q,q') \mid q,q' \in Q\}\\
 & \cup & \{\langle q,q_f\rangle \mid q \in Q, q_f \in F\}\\
 & \cup & \{\langle q\rangle \mid q \in Q\},\\
E & = & O - \{d\},\\
\Pi_1 & = & (O,[_{s_1}[_{1,2}\ ]_{1,2}\ ]_{s_1},q_0,\lambda,E,R_{s_1},R_{1,2}),\\
R_{s_1} & = & \{(q,out;q'a,in) \mid qa \rightarrow q' \in P\}\\
 & \cup & \{(q,out;\langle q'\rangle a,in) \mid qa \rightarrow q' \in P\},\\
R_{1,2} & = & \{(\langle q\rangle,in),\ (\langle q\rangle,out) \mid q \in Q\},
\end{array}$$

$$\begin{array}{rcl}
\Pi_2 & = & (O,[_{s_2}\ ]_{s_2},d,E,R_{s_2}),\\
R_{s_2} & = & \{(d,out;(q,q')a,in) \mid qa \rightarrow q' \in P, q \in Q\}\\
 & \cup & \{((q,q'),out;(q,q'')a,in) \mid q'a \rightarrow q'' \in P, q \in Q\}\\
 & \cup & \{((q,q'),out;\langle q,q_f\rangle a,in) \mid q'a \rightarrow q_f \in P, q \in Q, q_f \in F\},\\
R & = & \{(s_1,\langle q\rangle/\langle q,q_f\rangle,s_2) \mid q \in Q, q_f \in F\}.
\end{array}$$

The first component analyzes a prefix of a string in $L(A)$, the second component analyzes a suffix of a string in $L(A)$, first guessing a state $q \in Q$ from which the automaton starts its work. At some moment, the first component stops bringing objects inside by taking from the environment a symbol $\langle q'\rangle$ for some $q' \in Q$, reached after parsing the prefix of the string in $L(A)$. This object will pass repeatedly across the inner membrane of Π_1. The second component can stop if a state q' is reached in the automaton A for which no rule $q'a \rightarrow q''$ exists in P (and then Δ never stops, because its first component never stops), or after reaching a state in F, hence introducing an object of the form $\langle q,q_f\rangle$ for some $q_f \in F$. Note that q is the state chosen initially and always stored in the first position of objects (q_1,q_2) used by Π_2. The computation can halt only by using a communication rule from R, and this is possible only if $q = q'$ – the first component has reached the state of A which was the state from which the second component started its work. Consequently, the concatenation of the two strings introduced in the system by the two components is a string from $L(A)$. Thus, the language $L(A)$ is weakly parallelizable.

Now, consider a regular language such that no two adjacent symbols in a string can be permuted (take an arbitrary regular language L over an alphabet V and a morphism $h : V^* \longrightarrow (V \cup \{c\})^*$, where c is a symbol not in V, such that $h(a) = ac$ for each $a \in V$). Then, clearly, if the two strings accepted by the two components of the dP automaton Δ are of equal length (note that the strings of $h(L)$ are of an even length), then the time needed to Δ to accept the whole string is (about) half of the time needed to any P automaton Π which accepts the same language. This proves that the language $h(L)$ is efficiently parallelizable, hence we can state:

Theorem 3. *Each regular language is weakly ComX parallelizable, and there are efficiently ComX parallelizable regular languages, for all $X \in \{N, R, W\}$.*

Of course, faster dP automata can be constructed, if we use more than two components. However, it is not clear whether dP automata with $n+1$ components are always faster than dP automata with n components – this might depend on the structure of the considered language (remember that the distribution of the input string to the components of the dP automaton must be balanced). More specifically, we expect that there are (n,m) weakly parallelizable languages which are not, e.g., $(n+1,m)$ weakly parallelizable; similar results are expected for efficiently parallelizable languages.

A natural question is how much the result in Theorem 3 can be extended. For instance, is a similar result true for the linear languages, or for bigger families of languages? According to Theorem 2.3.5.4 in [9], this is not true for measures *ComR* and *ComW*, the recognition of context-free languages (actually, the language L_R at page 78 of [9] is linear) have already the highest communication complexity (in 2-party protocols), a linear one with respect to the length of the string. Thus, the number of communication rules used by a dP automaton during a computation cannot be bounded by a constant. The case of measure *ComN* remains to be settled: is it possible to have computations with a bounded number of communication steps, but with these steps using an unbounded number of rules? We conjecture that even in this case, languages of the form $\{x \ mi(x) \mid x \in \{a,b\}^*\}$, where $mi(x)$ is the mirror image of x (such a language is minimally linear, i.e., can be generated by a linear grammar with only one nonterminal), are not weakly *ComN* parallelizable.

Many other questions can be raised in this framework. For instance, we can consider families of languages: (n,m)-weakly *ComX* parallelizable, weakly *ComX* parallelizable, (n,m,k)-efficiently *ComX* parallelizable, and efficiently *ComX* parallelizable. Which are their properties: interrelationships and relationships with families in Chomsky hierarchy, closure and decidability properties, hierarchies on various parameters, characterizations and representations, etc.

Then, there is another possibility of interest, suggested already above: the static complexity measure defined as the cardinality of R, the set of communication rules. There is a substantial theory of descriptional complexity of (mainly context-free) grammars and languages, see [8], which suggests a lot of research questions starting from $ComS(\Delta) = card(R)$ (with “S" coming from “static") and extended to languages in the natural way ($ComS(L) = \min\{ComS(\Delta) \mid L = L(\Delta)\}$): hierarchies, decidability of various problems, the effect of operations with languages on their complexity, etc.

5 The Power of Controlling the Communication

In the previous sections the communication rules were used as any rule of the system, non-deterministically choosing the rules to be applied, with the communication rules competing for objects with the inner rules of the components, and observing the restriction of maximal parallelism. However, we can distinguish the two types of rules, “internal evolution rules" (transition rules, symport/antiport rules, rules with active membranes, etc.) and communication rules. Then, as in [1], we can apply the rules according to a priority relation, with priority for evolution rules, or with priority for communication rules. Moreover, we can place various types of controls on the communication channel itself. For instance, because the communication rules are antiport rules, we can associate with them promoters or inhibitors, as used in many places in membrane computing.

A still more natural regulation mechanism is to associate *states* with the channels, like in [7]. In this case, the communication rules associated with a pair (i,j) of components Π_i, Π_j are of the form $(q, u/v, q')$, where q, q' are elements of a given finite set Q of states; initially, the channel is assumed in a given state q_0. A rule as above is applied only if the cannel is in state q – and the antiport rule $(i, u/v, j)$ can be applied; after exchanging the multisets u, v among the two components Π_i, Π_j, the state of the channel is changed to q'.

An important decision should be made in what concerns the parallelism. In [7], the channel rules are used in the sequential mode, but we can also consider two types of parallelism: (i) choose a rule and use it as many times as made possible by the objects in the two components, or (ii) apply at the same time all rules of the form $(q,u/v,q')$ for various u and v (but with the same q and q'), in the non-deterministic maximally parallel way. In the result discussed below, any of these two possibilities works – and the result is somewhat surprising:

Theorem 4. *Any recursively enumerable language L is $(2,2)$-weakly ComN and ComR parallelizable and has $ComS(L) \leq 2$, with respect to extended dP automata with channel states.*

We do not formally prove this assertion, but we only describe the (rather complex, if we cover all details) construction of the suitable dP automaton.

Take a recursively enumerable language $L \subseteq T^+$, for some $T = \{a_1, a_2, \dots, a_n\}$. For each string $w \in T^+$, let $val_{n+1}(w)$ be the value of w when considered as a number in base $n+1$, using the digits $a_1, a_2, \dots, a_n$ interpreted as $1, 2, \dots, n$, without also using the digit zero. We extend the notation to languages, in the natural way: $val_{n+1}(L) = \{val_{n+1}(w) \mid w \in L\}$. Clearly, L is recursively enumerable if and only if $val_{n+1}(L)$ is recursively enumerable, and the passage from strings w to numbers $val_{n+1}(w)$ can be done in terms of P automata (extended symport/antiport P systems are universal, hence they can simulate any Turing machine; this is one of the places where we need to work with extended systems, as we need copies of a and b – see below – to express the values of strings, and such symbols should be taken from the environment without being included in the accepted strings).

Construct now a dP automaton Δ with two components, Π_1 and Π_2, working as follows. The component Π_1 receives as input a string $w_1 \in T^*$ and Π_2 receives as input a string $w_2 \in T^*$, such that $w_1 w_2$ should be checked whether or not it belongs to the language L. Without loss of generality, we may assume that $|w_1| \in \{|w_2|, |w_2|+1\}$ (we can choose a balanced distribution of the two halves of the string). In the beginning, the state of the channel between the two components is q_0.

Both components start to receive the input symbols, one in each time unit; the component Π_1 transforms the strings w_1 in $val_{n+1}(w_1)$ copies of a symbol a, and Π_2 transforms the string w_2 in $val_{n+1}(w_2)$ copies of a symbol b. When this computation is completed in Π_1, a special symbol, t, is introduced. For this symbol, we provide the communication rule $(q_0, t/\lambda, q_1)$, whose role is to change the state of the channel. We also consider the rule $(q_1, a/\lambda, q_2)$. Using it in the maximally parallel way, all symbols a from Π_1 are moved to Π_2, in one communication step.

Because we have considered w_1 at least of the length of w_2 and we also need two steps for “opening" the channel and for moving the symbols a across it, we are sure that in this moment in Π_2 we have, besides the $val_{n+1}(w_1)$ copies of a, $val_{n+1}(w_2)$ copies of b. The second component takes now these copies of a and b and computes $val_{n+1}(w_1 w_2)$, for instance, as the number of copies of an object c. After that, Π_2 checks whether or not $val_{n+1}(w_1 w_2) \in val_{n+1}(L)$. If the computation halts, then the string $w_1 w_2$ is accepted, it belongs to the language L.

Note that the dP automaton Δ contains two communication rules (hence $ComS(L) \leq 2$) and that each computation contains two communication steps (hence $ComN(L) \leq 2$), in each step only one rule being used (hence $ComR(L) \leq 2$). These observations complete the proof of the theorem.

Of course, $ComW(\Delta) = \infty$. (Similarly, if we define *ComR* taking into account the multiplicity of using the rules, then also *ComR* can be considered infinite – hence the assertion in the theorem remains to be stated only for the measure *ComN*.)

Instead of changing channel states as above, we can assume that the channel itself switches from “virtual" to “actual", like in population P systems, [3]: the channel is created by object t produced by Π_1, and then used for moving a from Π_1 to Π_2 by a usual communication rule (which, by definition, is used in the maximally parallel way).

Anyway, the conclusion of this discussion is that the results we obtain crucially depend on the ingredients we use when building our dP systems (as well as on the chosen definitions for complexity measures and types of parallelizability).

6 Closing Remarks

The paper proposes a rather natural way (using existing ingredients in membrane computing, bringing no new, on purpose invented, stuff into the stage) for solving problems in a “standard" distributed manner (i.e., splitting problems in parts, introducing them in various component “computers", and constructing the solution through the cooperation of these components) in the framework of membrane computing. So called dP schemes/systems were defined, and two notions of parallelizability were proposed and briefly investigated for the case of dP automata (accepting strings).

A lot of problems and research topics were suggested. The reader can imagine also further problems, for instance, transferring in this area notions and questions from the communication complexity theory, [9], considering other types of P systems (what about spiking neural P systems, where we have only one type of objects and no antiport-like rules for communicating among components?), maybe using unsynchronized P systems, non-linear balanced input, and so on and so forth. We are convinced that dP systems are worth investigating.

Note. During the recent Brainstorming Week on Membrane Computing, 1-5 of February 2010, Sevilla, Spain, several comments about the definitions and the results of this paper were made, especially by Erzsébet Csuhaj-Varú, György Vaszil, Rudolf Freund, and Marian Kögler. Several continuations of this paper are now in preparation; the interested reader is requested to check the bibliography from [17], in particular, the Brainstorming proceedings volume.

Acknowledgements

This work is supported by Proyecto de Excelencia con Investigador de Reconocida Valía, de la Junta de Andalucía, grant P08 – TIC 04200. Useful discussions with Henry Adorna are gratefully acknowledged.

Bibliography

[1] H. Adorna, Gh. Păun, M.J. Pérez-Jiménez: On communication complexity in evolution-communication P systems. Manuscript, 2009.

[2] L. Babai, P. Frankl, J. Simon: Complexity classes in communication complexity. *Proc. 27th Annual Symp. Founf. Computer Sci.*, 1986, 337–347.

[3] F. Bernardini, M. Gheorghe: Population P systems. *J. Universal Computer Sci.*, 10, 5 (2004), 509–539.

[4] M. Cardona, M.A. Colomer, A. Margalida, I. Pérez-Hurtado, M.J. Pérez-Jiménez, D. Sanuy: A P system based model of an ecosystem of some scavenger birds. *Membrane Computing. Proc. WMC10, Curtea de Argeş, 2009* (Gh. Păun et al., eds.), LNCS 5957, Springer, 2010, 182–195.

[5] M. Cavaliere: Evolution-communication P systems. *Membrane Computing. Proc. WMC 2002, Curtea de Argeş* (Gh. Păun et al., eds.), LNCS 2597, Springer, Berlin, 2003, 134–145.

[6] E. Csuhaj-Varjú: P automata. *Membrane Computing. Proc. WMC5, Milano, 2004* (G. Mauri et al., eds.), LNCS 3365, Springer, Berlin, 2005, 19–35.

[7] R. Freund, Gh. Păun, M.J. Pérez-Jiménez: Tissue-like P systems with channel-states. *Theoretical Computer Sci.*, 330, 1 (2005), 101–116.

[8] J. Gruska: Descriptional complexity of context-free languages. *Proc. Symp. on Mathematical Foundations of Computer Science, MFCS*, High Tatras, 1973, 71–83.

[9] J. Hromkovic: *Communication Complexity and Parallel Computing: The Application of Communication Complexity in Parallel Computing*. Springer, Berlin, 1997.

[10] M. Oswald: *P Automata*. PhD Thesis, TU Vienna, 2003.

[11] Gh. Păun: *Membrane Computing. An Introduction*. Springer, Berlin, 2002.

[12] Gh. Păun, G. Rozenberg, A. Salomaa, eds.: *Handbook of Membrane Computing*. Oxford University Press, 2010.

[13] M.J. Pérez-Jiménez: A computational complexity theory in membrane computing. *Membrane Computing. Proc. WMC10, Curtea de Argeş, 2009* (Gh. Păun et al., eds.), LNCS 5957, Springer, 2010, 125–148.

[14] A.E. Porreca, A. Leporati, G. Mauri, C. Zandron: Introducing a space complexity measure for P systems. *Intern. J. Computers, Communications and Control*, 4, 3 (2009), 301–310.

[15] G. Rozenberg, A. Salomaa, eds.: *Handbook of Formal Languages*. 3 volumes, Springer, Berlin, 1998.

[16] A.C. Yao: Some complexity questions related to distributed computing. *ACM Symposium on Theory of Computing*, 1979, 209–213.

[17] The P Systems Website: `www.ppage.psystems.eu`.

Gheorghe Păun graduated the Faculty of Mathematics, University of Bucharest, in 1974 and received his Ph.D. from the same university in 1977. From 1990 he is a senior researcher at the Institute of Mathematics of the Romanian Academy. He (repeatedly) visited numerous universities in Europe, Asia, and North America. His main research areas are formal language theory and its applications, computational linguistics, DNA computing, and membrane computing; this last research area was initiated by him, in 1998, and the respective models are now called P systems, see `http://ppage.psystems.eu`). He has published a large number of research papers, has lectured at many universities, and gave numerous invited talks at recognized international conferences. He has published eleven monographs in mathematics and computer science, has (co)edited over seventy collective volumes and special issues of journals, and also published many popular science books, books on recreational mathematics (games), and fiction books. He is a member of the editorial board of more than a dozen international journals and was/is involved in the program/steering/organizing committees for many recognized conferences and workshops. In 1997 he was elected a member of the Romanian Academy and from 2006 he is a member of Academia Europaea. He also got other honors, in Romania or abroad. He is an ISI Highly Cited Researcher (see `http://isihighlycited.com/`).

Mario J. Pérez-Jiménez received his degree in mathematics from the Barcelona University and the doctor degree in mathematics from the Sevilla University. In the past, he was associated professor at the University of Barcelona. Currently, he is full professor of Computer Science and Artificial Intelligence at the University of Sevilla, where is the head of the Research Group on Natural Computing. His main research fields are computational complexity theory, natural computing, membrane computing, bioinformatics, and computational modeling for systems biology and population biology. He has published twelve books in computer science and mathematics, and over 200 scientific papers in international journals (collaborating with many researchers worldwide). He is the main researcher in various European, Spanish and Andalusia research grants. He has been an independent expert to the evaluation of NEST (New and Emergent Science and Technology) proposals under the Sixth Framework Programme of the European Community, and from May 2006 he is an European Science Foundation peer reviewer.

Int. J. of Computers, Communications & Control, ISSN 1841-9836, E-ISSN 1841-9844
Vol. V (2010), No. 2, pp. 251-260

Implementing Web Services Using Java Technology

M. Pirnau

Mironela Pirnau
“Titu Maiorescu” University Bucharest
Romania, 22 Dambovnicului Street, District 4, 040051, Bucharest, Romania
E-mail: mironela.pirnau@utm.ro

Abstract:
There are several essential activities that need to take place in any service-oriented environment such as: a Web service has to be created, to have its interfaces and invocation methods defined, its service has to be published to one or more intranet or Internet repositories to locate potential users. A Web service needs to be located in order to invoke potential different users. The technologies offered by Web Services enable us to cast older applications, we can use them and the packages already existing in a certain enterprise. The infrastructure associated with older applications can also be wrapped as a serial set of services. In the following pages I tried to present general characteristics of the Web services up to present, as well as the usage of Java technologies for the implementation of web services. The present paper targets a systematisation of the main web services, as well as highlighting and testing the most important technologies used to develop them.
Keywords: Web Service, Java, XML, SOAP and Semantic Web.

1 Introduction

Web Services are software applications that conform to the Web Service Interoperability Organization. They are becoming more and more popular and reveal a model in which discrete tasks within e-business processes are widely distributed throughout a value net. The label for "web services" has two levels of meaning - one specific and one conceptual. Web services represent standards, which describe services oriented to applications with architecture having a base of components. These are software components, which can be reusable, having included functionality [11, 12, 16]. The Web Services use standard XML languages. They use HTTP for sending messages and they are independent of the platform and of the language. Web Services are freer systems, and the customer may not have knowledge of the Web Service up to the time it invokes, being suited to join requests of applications meeting demands of an Internet-wide [22].

2 Web Services Structure

Web Services systems maintain important decoupling and dynamic bonds of the elements. The entire elements in a system represent and publish an Application Programming Interface (API) message to different components in the network that work together. Services are comprised among applications by employing service discovery for the dynamic bonds of the partners. The Web Services show such architecture full of service orientated approach by orchestrating and finding out available network services, or even precise integration of the incidences that are based on the term of constructing applications.

The Web Services' architecture need three basic performs such as publish, find and bind. The providers publish services toward a certain service interloper [1, 4]. The requesters "find" demands services using an interloper and "bind" to them. The architecture of Web Services describes certain principles for building and also creating dynamic systems with no single involving.

There are many ways to instantiate a Web Service, choosing various implementation techniques for the roles and operations of the Web Services structure. From the structure of Web services we may mention: the service processes that involve more than one Web service, including discovery - belongs to this part of the architecture as it allows us to locate one particular service from among a collection of Web services. The architecture of Web Service invokes many layers and joins technologies [22] as we can see in the figure 1. Web Services can support other network protocols and distributed computing technologies. The implementation, according to the SOA principles, is based on the following protocols: HTTP, SOAP, WSDL, and UDDI [17].

HTTP[3] represents the most used Internet protocol and also the leading method of transmitting the information within the Web. Protocol of demand-answer type between the clients and the servers is another type of method to send data through Web. Simple Object Access Protocol (SOAP) is a format used to transmit XML message [1, 2]. It also represents a certain type of communication protocol among applications within Internet. It is a platform independent protocol no matter the language used and it permits the avoiding firewalls. Web Services Description Language (WSDL) is also a XML language that permits the specification of the accessing manner within the Web services. It allows the description of the web services, too.

Universal Description Discovery and Integration has the following acronym UDDI. It is a system of registers that is based on XML; it is independent of the platform. UDDI is requested by the SOAP messages that demand a web service [10]. It offers access towards the WSDL descriptive document within the web service, too.

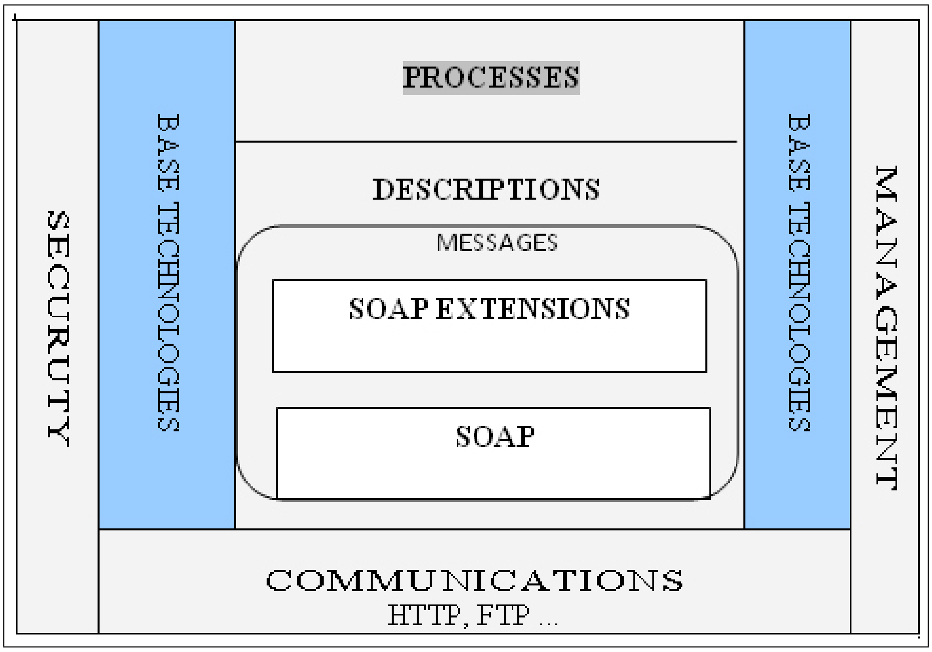
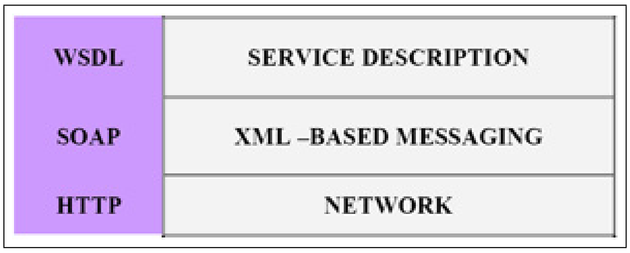


Figure 1: The architecture layers of the Web Services

The SOAP mainly specifies the manner we should format demands towards the server and the manner the server could build its own reply. SOAP represents a popular selection for Web Services. Along the transport, messages might be transmitted within a specific manner between servers and clients. The protocol of choice is represented by the HTTP architecture.

The pile depicted above (figure 2) enables Web Services to leverage the existing Internet infrastructure. It creates a low cost of entry to an omnipresent environment. The flexibility is not compromised by the interoperability requirement provided as alternative and value-add technologies [1]. An action that makes a WSDL document available is the service provider sending a WSDL document directly to a service requestor, named "direct publication". The service provider can publish the WSDL document describing the service to a host local WSDL registry, a private UDDI registry or to the UDDI operator.



WSDL	SERVICE DESCRIPTION
SOAP	XML –BASED MESSAGING
HTTP	NETWORK

Figure 2: The Interoperable base Web Services stack

3 Java Technologies

Web services represent web-based enterprise applications which use transport protocols that are XML- based, open standards in order to exchange data to their clients [18]. From the Java technologies we mention the following, presented in the table below [20].

Table 1: **Java Technologies**

WSIT	Web Services Interoperability Technology
JAX-WS10	The Java API for XML Web Services, Package javax.xml.ws
JAX-RPC	The Java API for XML Processing
JAXP	Provides a Java interface to DOM, SAX, and XSLT
JAXB	Java Architecture for XML Binding, Package javax.xml.bind
SAAJ	The SOAP with Attachments API for Java, Package javax.xml.soap
JAXR	The Java API for XML Registries
JSTL	JavaServer Pages Standard Tag Library
JSR 181	Java Web Services Metadata, Package javax.jws

As an example of these technologies I shall present the SAAJ [22] technology that stands for SOAP with attachments API for JAVA and that provides a standard set of APIs in order to send XML documents including attachments over the Internet. It is similar to JAX-RCC but it needs additional effort both on the client's and on the server's side [15, 21]. The SAAJ APIs provides a standard means for handling SOAP messages. It can be used to create, to inspect and to alter SOAP messages. SAAJ also contains methods for creating and sending attachments with the SOAP message. The attachments may contain data in the shape of any format, not being limited to XML as in the case of the SOAP messages. The attachments can be used to send inappropriate data in XML format as images for example. The SAAJ APIs offers the AttachmentPart class in order to represent the attachment part of a SOAP message [16]. Any SOAP Message object has automatically a SOAP Part object and requires sub-elements but because the AttachmentPart objects are optional, we must create and add them ourselves. SAAJ contains the API in order to create, populate and access SOAP messages, according to SOAP and SOAP Attachment specifications. SAAJ contains API needed to send non-provider-model messages [21]. We present the package relationship in figure 3 .

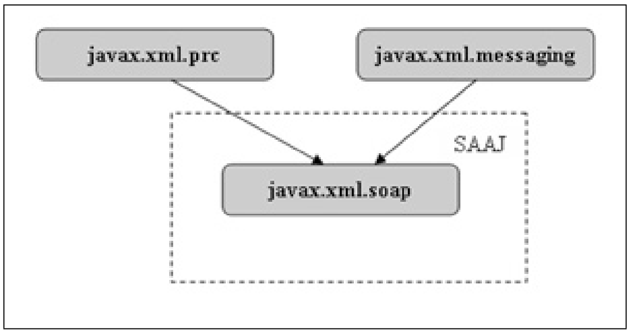


Figure 3: SAAJ - package relationship

SAAJ represents a part from JAXM (Java API for XML Messaging) which was evolved from the JSR expert group [1, 13, 21]. The final form of the specifications show two different aspects: the Core functionality concerned with manipulating of the SOAP messages and a very high level of messaging facility. Regarding SAAJ his an API of a very low standard, which is completely aware of the messages that are exchanged within the Web Service and its customers [16]. An application that uses SAAJ needs to construct SOAP messages little by little and take data out of the reply messages.

The usage of SAAJ demands a great deal of work deposed from the developer's side than JAX-RPC. We present a few cases in which it would be better to be helped by SAAJ and its relative JAXM instead of using JAX-RPC [1, 7].

JAX-RPC can be accessed just when the customer and the service are active at the same specific time and we also assume that there is a free network way to bond them. In the case the service is not free the moment the customer accesses it, there is no possibility of using JAX-RPC [17, 20].

JAXM is a service that provides a confident service able to deliver without need to demand the customer elements to be included in the way the confidence is offered. SAAJ as well as JAXM deliver a wide result disposal for messages based on XML.

The API from the **javax.xml.soap** package permits specifications such as: creating a piece of XML; creating access, adding, modifying parts of the SOAP message; creating a point-to-point contact to a certain endpoint; creating, adding, modifying a certain SOAP fault of information type etc. The table 2 includes the **javax.xml.soap** package and includes the API for building SOAP messages [4, 20]. The package is defined and included in SOAP with attachments API for JavaTM is named SAAJ.

SAAJ a tool for creating and sending messages

To allow sellers to bring up the implementations, the entire SAAJ API is built of interfaces and some classes that are abstract. In order to create a SOAP message, there is need to be used a MessageFactory type message. Part of the **javax.xml.soap** package represents an abstract class and that might be obtained by using the newInstance method. This type of method is represented by the public static MessageFactory newInstance that propagates SOAPException.

The Message Factory implies two methods to create the SOAP messages. The first method is represented by one public SOAPMessage createMessage that throws SOAPException. One public SOAPMessage createMessage(MimeHeaders headers, InputStream) which throw SOAPException. The second of the methods is used to non-serialize messages in a link of an input.

Table 2: **Methods included in the javax.xml.soap package**

Class Summary	
AttachmentPart	Single attachment to a SOAPMessage object
MessageFactory	Factory for creating SOAPMessage objects
MimeHeader	Object that stores a MIME header name and the value
SOAPConnection	Connection that a client might use for transmitting messages to a certain remote party (URL)
SOAPConnectionFactory	Factory for creating SOAPConnection objects
SOAPElementFactory	Creating SOAPElements
SOAPFactory	Creating various objects existing in the SOAP XML stack
SOAPMessage	SOAP messages root class

The default messages that comes back by the MessageFactory createMessage method contains a SOAPPart but also contains no types of attachments. An object of SOAPEnvelope style is included in the SOAPPart. In the middle of the envelope there are certain parts such as the header, made by an object able to implement the SOAPHeader interface and a SOAP message body in the shape of an object of SOAPBody classification. The body and the header, both contain certain XML elements. The Java technologies represent methods for constructing web services [8].

4 Related Work

The Web Semantic implemented and proposed use of the machinery to extend the Web represents a common goal. The multiple demands on the Web create a semantically temptation having a highly expressive formalisms because the Web represents a principled architecture of standards, formalisms and languages [6, 14].

What is specific for Web 2.0 is that it connects humans among them and for Web 3.0 it is specific that it connects humans to information in a very innovative new manner.

Web 3.0 functions as an intelligent assistant, which perceives the concepts of the Web pages and transmits important information of one individual. It not only looks for robotically through keywords. Tim Berners-Lee is the person who founded the World Wide Web. He defines the future online environment as being the Semantic

Web [5, 8, 9]. This is the same Web 3.0 a type of Internet that perceives information and not just transmits it to the client. Web 3.0 will be capable to offer for visitors complete data on any subject. The communities that are based on the same complete data exchange will certain develop. Mobile Web which is supported by mobile intelligent terminals would be able to enable the data access everywhere and anytime.

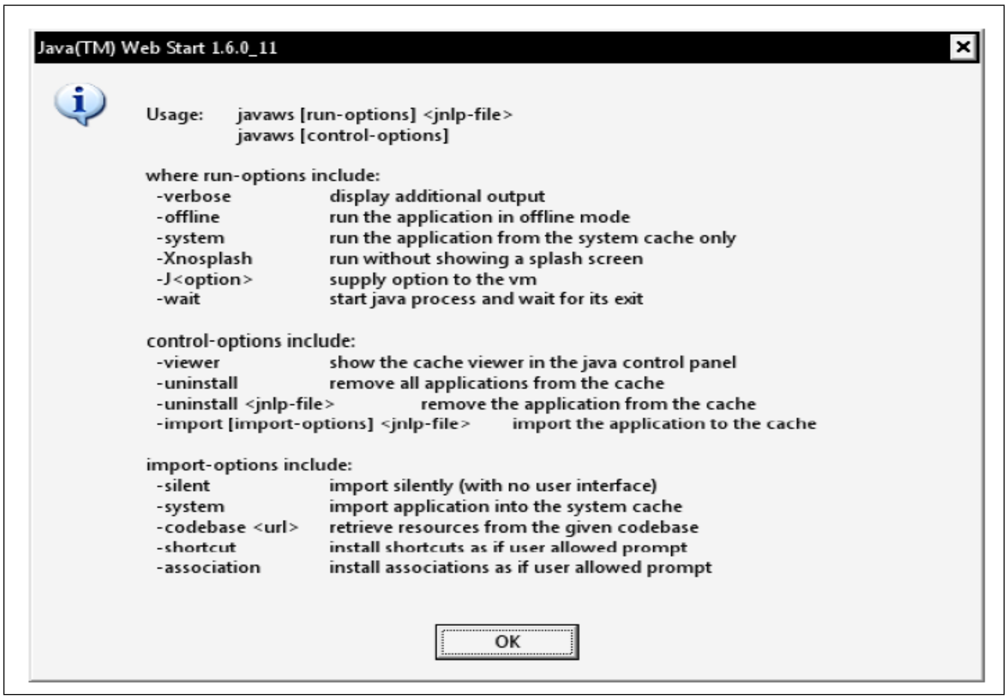
Rich Internet Applications (RIA) would appear more and more. Web 3.0 represents a much more direct, closer, measurable, reliable and honest communication.

The Metro style is made up from: JAX-WS, JAXB, and WSIT and within it is possible to create Web services that are secure, transactional, confident and interoperable, also certain customers [15]. The Metro elements are a certain part of the Project Metro, which part of GlassFish, Java EE (Java Platform, Enterprise Edition), and only part in Java SE (Java Platform, Standard Edition). Both GlassFish and Java EE represent support for JAX-RPC API [15, 17].

5 Web Applications. Java Web Start Technology

JWS (Java Web Start) represents a technology for installing, launching, and updating the java applications directly from the Web and uses the JNLP (Java Network Launching Protocol). For Java applications, that use of the JWS technology launches a range of order of javaws application as the following syntax:

javaws [options] [URL], and the options are those in figure 4.



```
Usage:    javaws [run-options] <jnlp-file>
          javaws [control-options]

where run-options include:
  -verbose              display additional output
  -offline              run the application in offline mode
  -system               run the application from the system cache only
  -Xnosplash            run without showing a splash screen
  -J<option>            supply option to the vm
  -wait                 start java process and wait for its exit

control-options include:
  -viewer               show the cache viewer in the java control panel
  -uninstall            remove all applications from the cache
  -uninstall <jnlp-file>          remove the application from the cache
  -import [import-options] <jnlp-file>      import the application to the cache

import-options include:
  -silent               import silently (with no user interface)
  -system               import application into the system cache
  -codebase <url>       retrieve resources from the given codebase
  -shortcut             install shortcuts as if user allowed prompt
  -association          install associations as if user allowed prompt
```

Figure 4: Options for Java Web Start

The role of the programmer: creates a Java application and archives the program along with its all resources; signs the jar archive (if necessary) and creates a configuration folder (jnlp) that describes the application; creates a Web page having a link to the configuration folder and copies on the web server, the **jar** folder, the **jnlp** and **Html** one.

The role of the user: accesses the link to the application from the programmer's Web page; the application will be installed automatically, locally and installs a JRE if necessary; launches in execution the application; for each execution of the application, it verifies if a new version on the programmer's new web page exists.

As a practical application, I have created a project that contains Java classes with administration role of the students. The purpose of the application is to use the Java Start technology and to realize "a following" in using the application for this technology as well as other technologies that I have tested (for example the Servlets technology etc). To realise the source, I also used the classes from the **Swing, AWT, SQL** etc packages. If we open the application in the NetBeans IDE 6.7 [23] environment, the project window looks like the one in figure 5.

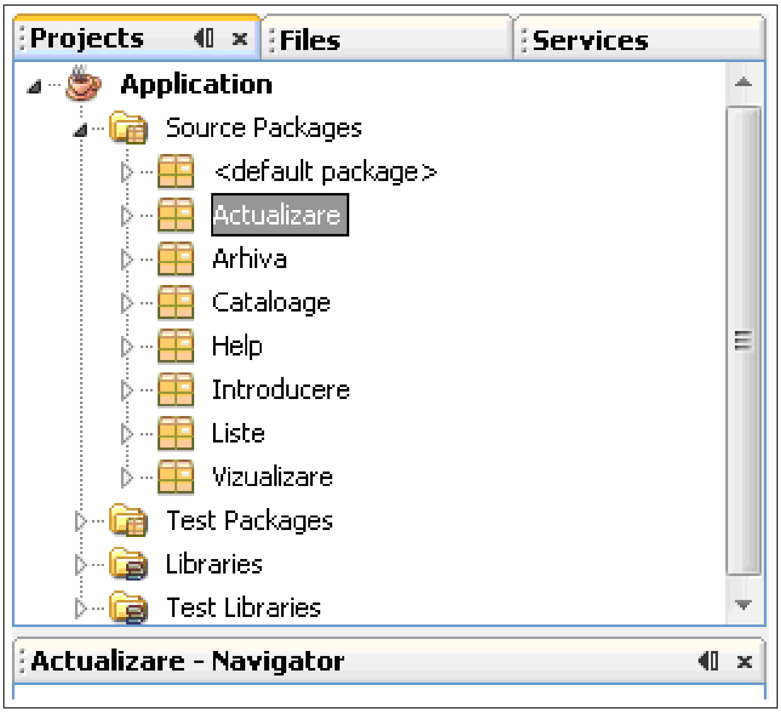


Figure 5: The window of the application opened in Netbeans

Helped by Java Web Start, the users may access one Java application only with click of a mouse on an HTML link towards a JNLP file for the certain applications from inside the web browser [19].

This should be installed onto the customer machine. It is not necessary to install JDK (Java Development Kit). In order to permit our Java application to be ran with Java Web Start, we have to configure the main properties of the manner the IDE could construct the project. The moment Java Web Start is accessed, the IDE creates automatically a JNLP file and an HTML page along with the link to the JNLP file, along with the JAR file. For the beginning, we must configure the project in order to activate Java Web Start and to locally test the application. So we must choose the Properties option from the project's shortcut menu, and in the window from the figure 6 the suitable settings are set.

The application's project must be configured so that the Java Web Start technology may be used at the launching.Certain sources are mixed together and then the main class of the project is launched in execution. The IDE mixes sources and we might see the Java screen starts, and in the warning window we are asked if the application might be executed and also to Select the checkbox and click Run inside the proper warning window.

The *"Application"* application begins, the same as in figure 7.

After activating the application, the **Dist** folder is created and contains for Java Web Start, the following two additional folders:

- launch.jnlp

It is an XML file having special parts and specifications instructing browsers how to run the application. The attributes for the JNLP files might include the JNLP version, application title, the seller's name, a link of the application of the JAR file etc. The configuration folder *jnlp* is in an XML format and offers information regarding the application that will be installed using JWS. Its structure is presented in figure 8.

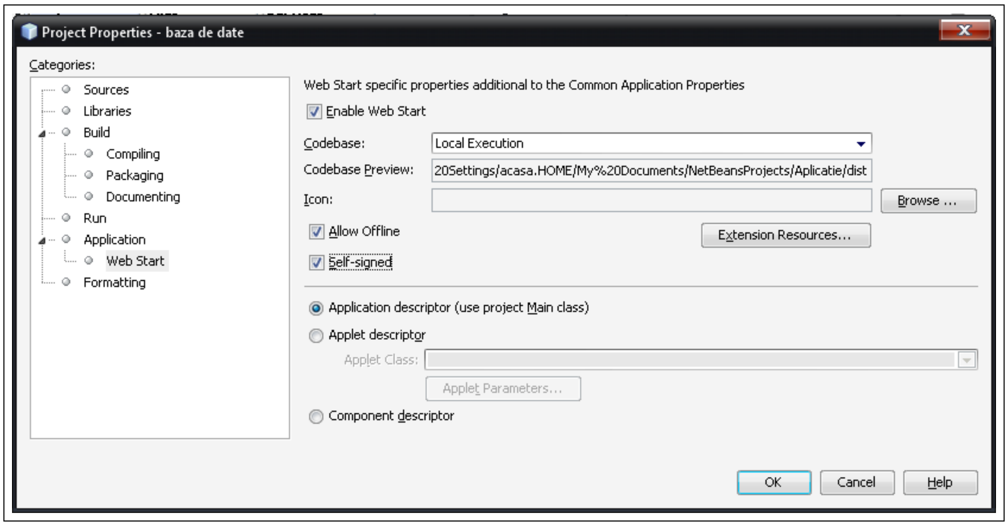
- launch.html

It is generated in an automatic HTML page a link to the JNLP file. The users might click the link in order to begin applications through Java Web Start (figure 9).

How to Run an Application from a Distant Location

The moment we are sure that the application begins in a successfully manner using Java Web Start from local sources, we might upload this to a distant location and start launching it from far there.

In order to develop applications using Java Web Start inside the web, the server of the Web is able to manipulate JNLP files. The web server can be configured in order to recognize JNLP files as applications, such as the MIME type for JNLP. Each Web server has a certain manner for adding MIME types. To develop our application to the



Web Start specific properties additional to the Common Application Properties

☑ Enable Web Start

Codebase: Local Execution

Codebase Preview: 20Settings/acasa.HOME/My%20Documents/NetBeansProjects/Aplicatie/dist

Icon:

☑ Allow Offline

☑ Self-signed

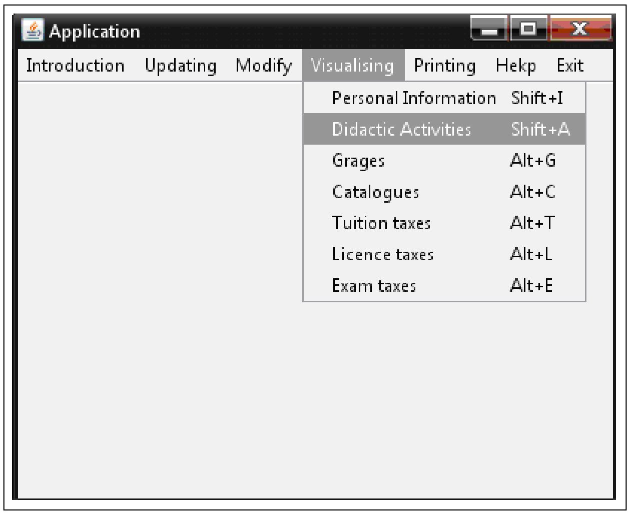
● Application descriptor (use project Main class)

○ Applet descriptor

Applet Class:

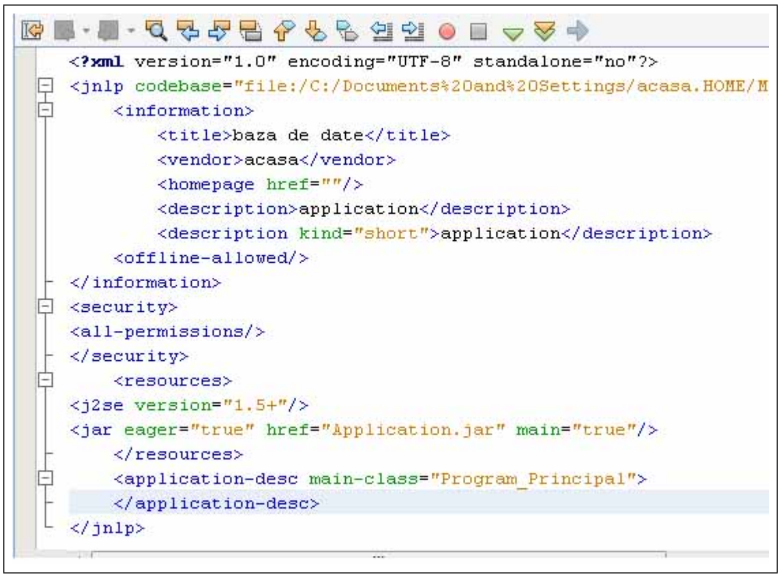
○ Component descriptor

Figure 6: Project Properties Web Start



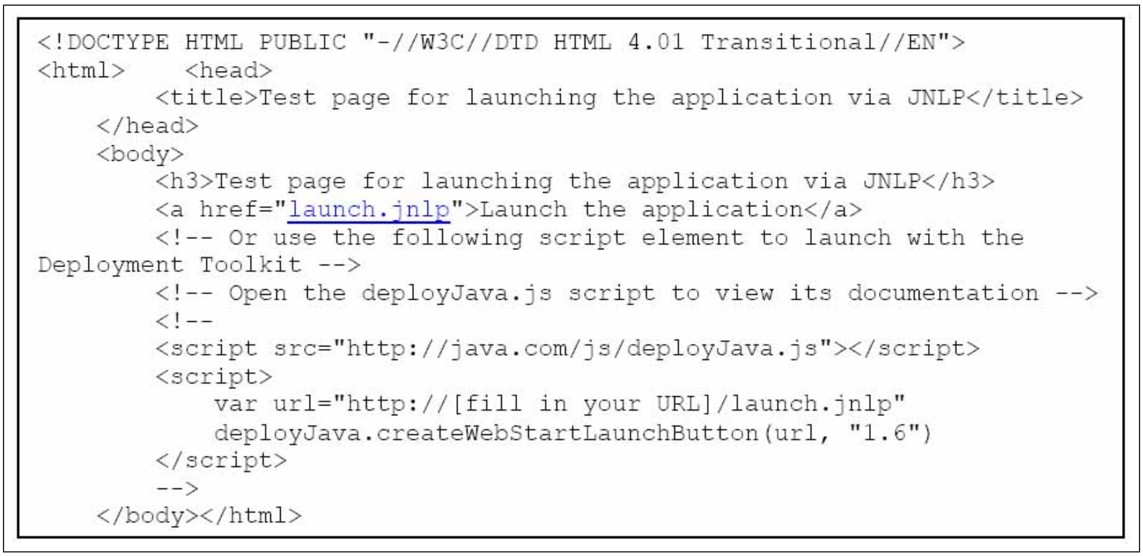
Personal Information	Shift+I
Didactic Activities	Shift+A
Grages	Alt+G
Catalogues	Alt+C
Tuition taxes	Alt+T
Licence taxes	Alt+L
Exam taxes	Alt+E

Figure 7: The main menu of the application



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<jnlp codebase="file:/C:/Documents%20and%20Settings/acasa.HOME/M
    <information>
        <title>baza de date</title>
        <vendor>acasa</vendor>
        <homepage href=""/>
        <description>application</description>
        <description kind="short">application</description>
    <offline-allowed/>
</information>
<security>
<all-permissions/>
</security>
    <resources>
<j2se version="1.5+"/>
<jar eager="true" href="Application.jar" main="true"/>
    </resources>
    <application-desc main-class="Program_Principal">
    </application-desc>
</jnlp>
```

Figure 8: The structure of the **launch.jnlp** file

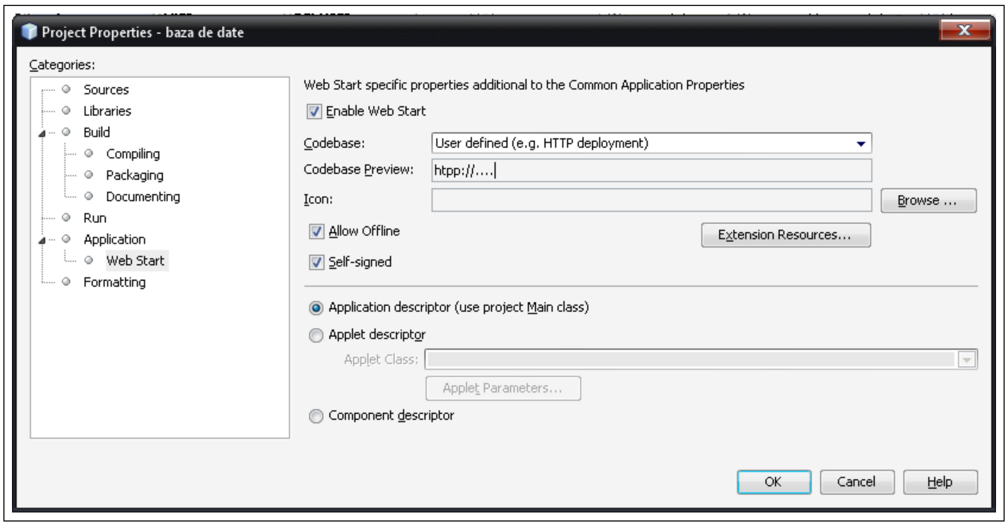


```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>    <head>
        <title>Test page for launching the application via JNLP</title>
    </head>
    <body>
        <h3>Test page for launching the application via JNLP</h3>
        <a href="launch.jnlp">Launch the application</a>
        <!-- Or use the following script element to launch with the
Deployment Toolkit -->
        <!-- Open the deployJava.js script to view its documentation -->
        <!--
        <script src="http://java.com/js/deployJava.js"></script>
        <script>
            var url="http://[fill in your URL]/launch.jnlp"
            deployJava.createWebStartLaunchButton(url, "1.6")
        </script>
        -->
    </body></html>
```

Figure 9: The structure of the **launch.html** file

customer, we should be sure that a Web server accesses all the files that contain our application. This may represent typically amounts for copying one or many of the JAR files, with a JNLP file, to the directories of the Web server.

The settings needed to enable the Web site sustain that Java Web Start is quite the same to develop a content based on HTML. In order to launch the application out of the web, we have to use a certain link of the files of applications source in the JNLP file. Regarding this, the project is chosen and in the **"Properties"** window the **CodeBase** section is defined. The URL where the files are loaded has the JNLP extension and also the jar and *Html* extension (figure 10).



Project Properties - baza de date

Web Start specific properties additional to the Common Application Properties

☑ Enable Web Start

Codebase: User defined (e.g. HTTP deployment)

Codebase Preview: htpp://....

Icon:

☑ Allow Offline

☑ Self-signed

● Application descriptor (use project Main class)

○ Applet descriptor

Applet Class:

○ Component descriptor

Figure 10: Settings for running applications on the Web

At this moment we may run our application, in the window of a browser, we type the URL towards the launch.html file and then press the "Launch the application" connection. Our "Application" application begins with Java Web Start. After analyzing the current application, we may observe that the integration of the applications and of the data sources must be realized without significant changes of the applications and of the data. Because the Web allows the easy access to information and services, it became a main element of communication having the condition to be used for applications that run it, adequate technologies such as those implemented on the Java platform.

6 Conclusions

The Web services represent a wide range of technologies used in the application of distributed implementation at the web system level. These technologies develop a program-to-program communication pattern using the XML standard, which assures a neutral shape report to platforms, programming languages and software specifications.

A Web services application is an application that interacts with the world using XML for data definition, WDSL for service definition and SOAP for communication with another software. The fundamental idea of the web services is the integration [16]. This concept represents a set of standard technologies that facilitate the interoperability among heterogenic systems at an organizational level or on the Internet.

For the optimization of the Web aplications, the companies head more and more to the programming environment of Java and of the relevant platforms. Java represents an independent language regarding the platform that is projected for the Web and, at the same time, it is a developing platform of various aplications, no matter the company's specific. Next to the Java platform the XML technology has developed having the goal to exchange data among the aplications.

As future research I intend to run more applications on Java different platforms and to compare both the manner of using these for working in the environment and the security level implemented the platforms.

Bibliography

[1] E. Hewitt, *Java SOA Cookbook*, O'Reilly, 2009.

[2] J. Governor, D Hinchcliffe, D , *Web 2.0 Architectures*, O'Reilly, 2009.

[3] J. Zukowski, *Java 6 Platform Revealed*, SpringerLink Date, 2006.

[4] *** , *SOAP with Attachments API for Java (SAAJ)*, Sun Microsystems, Tech. Rep., 2004.

[5] R. Volz, S. Handschuh, S. Staab, L. Stojanovic, and N. Stojanovic, Unveiling the Hidden Bride: Deep Annotation for Mapping and Migrating Legacy Data to the Semantic Web, *J. Web Semantics*, vol. 1, no. 2, pp. 187-206, 2004

[6] I. Dzitac, B. E. Barbat, Artificial Intelligence + Distributed Systems = Agents, *International Journal of Computers, Communications and Control*, 4(1):17-26, 2009.

[7] R. Chinnici, *Java API for XML-Based RPC (JAXRPC)*, Java Community Process, Tech. Rep., 2003.

[8] R. Nagappan, R. Skoczylas, R. Sriganesh, *Developing Java Web Services*, Wiley 2003.

[9] I. Dzitac, S. Dzitac, E. Valeanu, Web Distributed Computing for Landscape Architecture, in Eleftheriadis, N., Styliadis, A., Paliokas, I. (eds), *Proceedings of the LANT07, International Conference Landscape Architecture and New Technologies*, 25-26 May 2007, Drama, Greece, pp. 25-36, 2007.

[10] ***, *Java 2 Platform, Enterprise Edition Specification*, Sun Microsystems, Tech. Rep., 2003;

[11] L. Duta, F.G. Filip, Control and decision making in disassembling used electronic products, *Studies in Informatics and Control*, 17 (1):17-28, 2008.

[12] L.A. Zadeh, D. Tufis, F.G. Filip, I. Dzitac (Eds), *From Natural Language to Soft Computing: New Paradigms in Artificial Intelligence*, Editing House of the Romanain Academy, Bucharest, 2008.

[13] N. Kassem, A. Vijendran, and Rajiv.Mordani, *Java API for XML Messaging (JAXM)*, Sun Microsystems, Tech. Rep., 2003, http://java.sun.com/xml/downloads/jaxm.html.

[14] http://www.nowpublishers.com/.

[15] http://java.sun.com/webservices/technologies/index.jsp.

[16] http://en.wikipedia.org/wiki/Service-oriented_architecture.

[17] http://java.sun.com/developer/technicalArticles/WebServices/soa/.

[18] http://java.sun.com/javase/technologies/desktop/javawebstart/index.jsp.

[19] http://java.sun.com/javase/6/docs.

[20] http://java.sun.com/webservices/docs/1.6/tutorial/doc/.

[21] http://www.javapassion.com/webservices.

[22] http://www.w3.org/TR/ws-arch/.

[23] http://www.netbeans.org.

Mironela Pirnau (1966) has a M.Sc. in Automatics, Faculty of Automatics and Computing, Politechnical University in Bucharest (1992) and PhD in Electronics (2003) from University of Pitesti. Her current interests include Applied Informatics. Since 2008 she has been a Lecturer PhD at the Faculty of Science and Technology of Information, Titu Maiorescu University in Bucharest. She has published books and articles on Informatics. She has recently been interested in researches on the use of Java technologies in developing applications of Informatics.

Author index