ITHEA

International Journal

INFORMATION TECHNOLOGIES & KNOWLEDGE

2015 Volume 9 Number 3

International Journal
INFORMATION TECHNOLOGIES & KNOWLEDGE

Volume 9 / 2015, Number 3

**International Journal “INFORMATION TECHNOLOGIES & KNOWLEDGE” (IJ ITK)
is official publisher of the scientific papers of the members of
the ITHEA International Scientific Society**

IJ ITK rules for preparing the manuscripts are compulsory.
The **rules for the papers** for IJ ITK are given on www.ithea.org

Responsibility for papers published in IJ ITK belongs to authors.

International Journal “INFORMATION TECHNOLOGIES & KNOWLEDGE” Volume 9, Number 3, 2015
Edited by the **Institute of Information Theories and Applications FOI ITHEA**, Bulgaria, in collaboration with:
Institute of Mathematics and Informatics, BAS, Bulgaria; V.M.Glushkov Institute of Cybernetics of NAS, Ukraine;
Universidad Politechnika de Madrid, Spain; Hasselt University, Belgium;
St. Petersburg Institute of Informatics, RAS, Russia; Institute for Informatics and Automation Problems, NAS of the Republic of Armenia.
Printed in Bulgaria
Publisher ITHEA®
Sofia, 1000, P.O.B. 775, Bulgaria. www.ithea.org, e-mail: info@foibg.com
Technical editor: Ina Markova

Издател: ИТЕА®, София 1000, ПК 775, България, www.ithea.org, e-mail: info@foibg.com

ISSN 1313-0455 (printed) **ISSN 1313-048X (online)**

USER-CENTRIC AND CONTEXT-AWARE ABC&S

Ivan Ganchev

***Abstract**: The evolving Always Best Connected and best Served (ABC&S) communication paradigm is addressed in this paper. The goal is to propose aspects of a novel vision together with the consequential strategic requirements and potential solutions for system architectural development. Subjective and objective aspects of the ABC&S concept from the viewpoints of the various stakeholders – mobile users, access network providers (ANPs), mobile service providers (xSPs), and mobile device manufacturers – are delved into. As primarily ABC&S is an end-user's issue this perspective is given priority. The importance of context awareness is highlighted and the main context types are identified. An illustrative ABC&S example is presented along with corresponding IT solutions.*

***Keywords**: Always Best Connected and best Served (ABC&S), user centricity, context awareness, mobile app, cloud-based system, service recommendations, Ubiquitous Consumer Wireless World (UCWW).*

***ACM Classification Keywords**: H.3.4 Systems and Software – User profiles and alert services, C.2.1 Network Architecture and Design – Wireless communication, D.2.2 Design Tools and Techniques – User interfaces, H.1.2 User/Machine Systems – Human information processing.*

1. Introduction

The "Always Best Connected" (ABC) path of the evolution of the future communications world was proposed in 2003 [Gustafsson, 2003], [O'Droma, 2003]. The ABC concept was later extended to an "Always Best Connected and best Served" (ABC&S) paradigm [O'Droma, 2006], [Ganchev, 2007]. As part of the ABC&S functionality, in general, users evaluate their options in relation to the provision of desired mobile services through access networks available to them, and make decisions on which (best) access network to use, under what conditions and for which particular service instance.

The ABC&S concept may be considered as the vision where logical telecommunication connections, needed for services being used by mobile users, are realized in a way that would be regarded as 'best' by these users [O'Droma, 2006]. What is 'best' varies widely according to the viewpoint of the stakeholders involved, the user's location, the network environment enveloping the user, the service instances being provided and the time the service is being accessed (time of day, day of week, etc.).

With this kind of general descriptive definition, what ABC&S actually would be at any time, in any place, for any user, mobile service provider (xSP), and access network provider (ANP) will vary and will evolve radically as the full gambit of future wireless networks themselves evolve.

In future wireless networks, the ABC&S vision should also include the capability of flexible management of the all-important quality of service (QoS) requirements ranging over all protocol layers [O'Droma, 2004]. In environments containing multiple wireless (and fixed) networks, this vision includes open access to all of these networks, under certain conditions, and the ability to create, and update, a mix and match of desired service instances based on acceptable price/performance ratios – all responding to requirements set down by users in their profiles. Whether in a mobile or fixed context, it also includes a capacity to advertise, discover and learn about new networks (and services), new network- and service options and price/performance offerings, and dynamically change access, without losing service session, in accordance with user preferences, or ANP's, or xSP's obligations to meet per-service or per-connection service level agreements (SLA), or service dependent xSP-desired QoS performance. Largely the above is coming at an ABC&S definition from a user's viewpoint. There are other viewpoints – e.g. the mobile service providers', the access network providers' and the mobile device manufacturers' viewpoints, presented in the next section.

In many ways the evolution of mobile communications envisages a stage being reached where the competition among ANPs may be a consideration for each user call/session [O'Droma, 2004]. For example, whenever a phone call is to be made, if the user is within the footprint of several access networks, then clearly s/he would like, in a painless user-friendly way, to choose the network which currently offers the 'best' or most acceptable price/performance ratio for the service being sought. Important also to the user is the need to control the billing options available at any time. This presents a user requirement for a different type of approach to the traditional authentication, authorization and accounting (AAA) structure, as it mitigates against a user's being tied (i.e. as a subscriber) to one ANP – the home access network provider. It may facilitate the user better for the AAA functionality to be mediated by a third party (or parties) [McEvoy, 2005], [Tairov, 2011]. For instance for the traditional service of making a phone call, a user might want different calls to go through different ANPs (3G, 4G, Wi-Fi, Wi-MAX), being selected based on the particular callee, the day of the week, the time of the day, and other user profile's characteristics such as the user role – professional, personal, etc. To do this, mobile devices could have a mobile ABC&S application installed, through which all outgoing phone calls could be easily made. This option is presented in detail in Section 4.

In some cases the user may prefer not 'the best' connection (with best cost/performance ratio) for a given service, but another connection not so expensive and providing satisfactory QoS (for this user in their present or specified role), [O'Droma, 2004]. For this purpose the cost/performance ratio must be present in suitable measures understandable by the user, e.g. if s/he chooses not ‘the best’ connection, the user should know how much cheaper it is and what the likely corresponding drop of performance/QoS is.

In the case of using a multi-access device and if the user wants to use two or more services simultaneously (e.g. speaking on the phone and at the same time downloading and reading e-mails with attachments, and also browsing the web) and if s/he is within an environment of overlapping footprints of multi-access wireless networks in the current location, then the user’s mobile device may propose using different connections via different ANPs – sorted in descending order according to their cost/performance ratio – for different services, e.g. 4G for a business-type phone conversation and Wi-Fi for e-mail downloading and web browsing [O’Droma, 2004].

As indicated in [O’Droma, 2004], a real drive towards an open ABC&S paradigm has the potential to gradually restructure the existing subscriber-based and network-centric business realization of mobile communications, transforming it into a consumer-centric one. In this, it raises important challenges for existing ANPs and opens new opportunities for new ANPs, aiming to fill niche markets. For the latter it would mean the possibility of ease of entry and of having dynamic (and even casual) consumerist-like relationships with users, i.e. offering and providing services without any prior business relationship and subscription with them, which is realizable through utilization of a 3P-AAA mechanism [McEvoy, 2005], [Tairov, 2011].

For the mobile user the experience of ABC&S communications services should preferable move towards having consumerist-type characteristics where, for instance, through user-friendly interfaces ABC&S decisions are user-driven and user-executed. The most advanced ABC&S scenarios should enable users to move seamlessly between different (wireless) access networks according to their own criteria, e.g. on the basis of a comparison of the price/performance profiles of the networks, while maintaining active/on-going service sessions, i.e. without needing to reinitiate a session, or restart an application. The user-centric and user-driven ABC&S paradigm realization leads to a ubiquitous consumer wireless world (UCWW) communication environment [O’Droma, 2007], [O’Droma, 2010], where connectivity is available anywhere-anytime-anyhow, mobile services are rapidly deployed on-demand, customized to the user’s needs, and adapted to the current context in the best possible way independent of the user’s movement across heterogeneous access networks. This vision requires unprecedented levels of autonomy, service adaptability, and network element integration at all levels including device equipment, access networks, and mobile services, [O’Droma, 2004].

Most ABC&S decisions, and probably the easiest from an implementation viewpoint, will be made by the user on the basis of criteria many of which will be set down in multidimensional user-, device-, network-, and service profiles [O'Droma, 2004]. The range and sophistication of such profiles will grow with time in parallel with the growth in the range of device, network and service access options and will consist of complex sets or arrays of competing parameters. How this may be managed in a dynamic adaptable way is not a small challenge. An important research and development (R&D) goal here is the finding of solutions for automation of the entire process of advertisement, discovery, request, association, configuration and use of access networks and mobile services so that the user will not only be served anywhere-anytime-anyhow but also always be best served.

In this paper, the focus is on the ABC&S user centricity and context awareness. The goal of the former is to place wide-ranging freedom and control in the user's hands as regards access networks' and mobile services' choices, based on personal ABC&S criteria such as price/performance ratio matched to the user profile. The goal of the latter is to take into account the current context (user-, network-, service context) in order to make informative ABC&S decisions.

The rest of the paper is organized as follows. Section 2 presents the ABC&S viewpoints of the main stakeholders. Section 3 deals with ABC&S related context. Section 4 describes an illustrative ABC&S example along with corresponding IT solutions. Finally, section 5 concludes the paper and presents future directions for research.

2. ABC&S viewpoints

ABC&S encompasses a vision, which may be defined differently by different stakeholders as outlined in [O'Droma, 2006]. Interpretations and viewpoints vary as a function of the 'interest' of the stakeholder. The definition of criteria for 'better' and 'best' consists of objective and subjective aspects. Normally not only will the categories of stakeholders such as users, ANPs, xSPs and mobile device manufacturers represent broad classes of expectations and requirements of what ABC&S is, but there will also be a wide range of divergent viewpoints within each category related to such matters as socio-politico-economic and regulatory environments, user population densities, service specialization, and geographic/territorial environments [O'Droma, 2006].

In so far as ABC&S has been considered, it has been in the contexts, where the service choice is usually being made by the user according to the perceived cost/performance ratio. However in general, as the ABC&S environment evolves, the user may not be the sole decision maker. ANPs and xSPs also have an interest, more or less keen depending on the circumstances. Besides the cost issue, other items of performance for consideration by all these interested parties include network-related QoS criteria (e.g. bandwidth, jitter, delay, packet loss, network congestion level, etc.), security requirements, subscriber/user loyalty, service performance history (e.g. response time, reliability, etc.), service provider↔user cost sharing model, etc. [O'Droma, 2004].

A good overview of different ABC&S viewpoints is provided in [O'Droma, 2006] and is summarized in Table 1.

With ABC&S decision-making process being user-driven and/or xSP-driven, competition within and among ANPs should lead to a suitably wide variety of bearer service products with appropriate price/performance configurations [O'Droma, 2004]. Internally for ANP, defining and adapting these price/performance configurations, and keeping users informed about the latest offerings will be a challenging exercise.

Table 1. A summary of the ABC&S viewpoints of the main stakeholders.

	Access Network Providers (ANPs)	Mobile Users	Mobile Service Providers (xSPs)	Mobile Device Manufacturers
Type of control	Centralized with most control being in their hands	Decentralized	Decentralized but supported by ANPs	Centralized or decentralized
Scope	Mostly within their domain only	Global	Global	Global
Driven by whom?	ANPs	Users (and perhaps in cooperation with xSPs)	xSPs in cooperation with users	ANPs, or users, or xSPs.
Goal	More bandwidth, improved range and quality of coverage, attractive tariff plans.	Open consumerist-type price/performance service offering with comprehensive service access	Flexibility in development and deployment of services through third-party networks via open interfaces, attractive pricing mechanisms.	Intuitive GUIs, wide range of network interfaces, greater device reconfigurability options.

3. ABC&S related context

The ABC&S decisions depend greatly on the current context. Besides the context that relates to the mobile *services* available on offer (i.e. the category, type, scope and attributes of the service, the request time, the application initiating the request, the current QoS/QoE index of the service component, price, etc.), the context data may relate to the *user* (e.g., the user location, local time, weather, environmental state, current battery charge and other operational characteristics of the user's mobile device, the user preferences, type of activity, intentions, engagements, social interests, the upper bound on the price and the lower bound on QoS accepted by the user for each particular service, privacy and security requirements, etc.), and/or relate to the constraints of the wireless access *network* currently utilized by the user (e.g., the communication channel state information (CSI), network congestion level, the current data usage pattern, the current QoS/QoE index, the cost of using the network, pricing scheme, etc.). Then determining the 'best' service instance at any moment for a particular user is based on a set of context parameter values, categorized in three groups – user-related *(u)*, service-related *(s)*, and (access) network-related *(n)*, forming a 3D *(u,s,n)* context space, as illustrated in Figure 1.

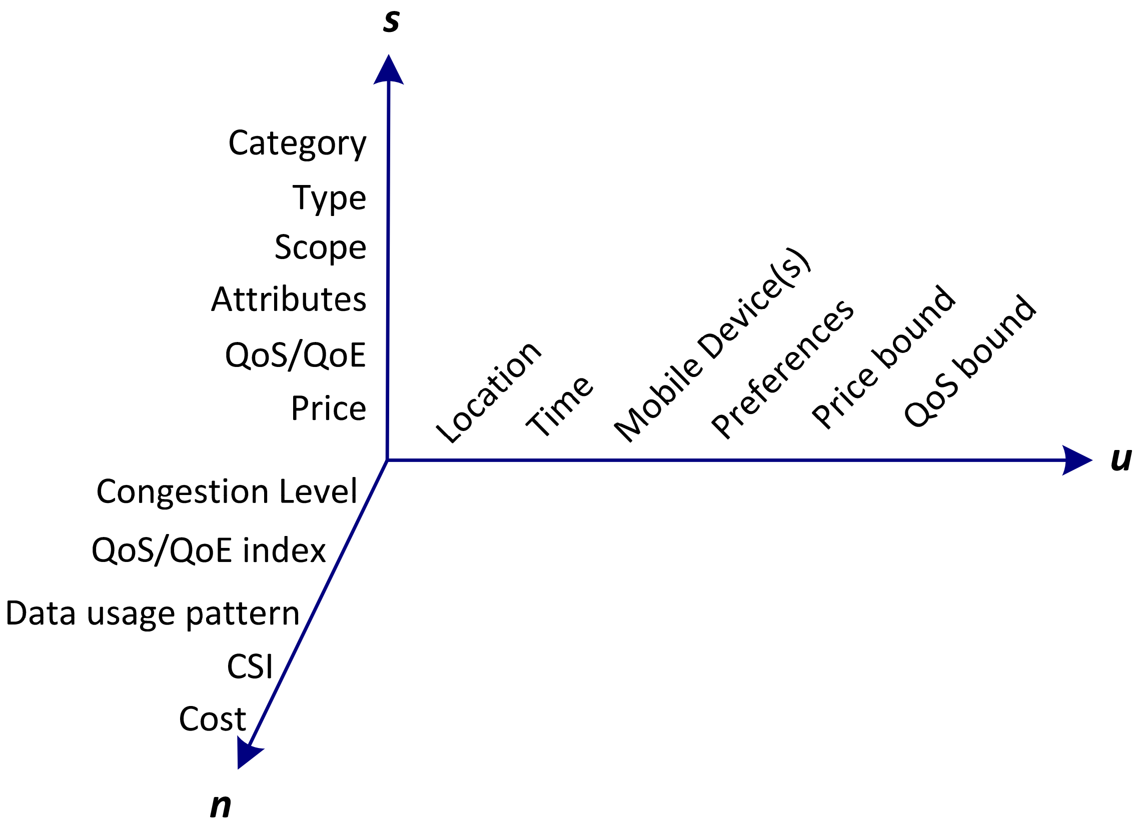


Figure 1. The 3D context space.

The concept of context allows making smart decisions based on mining of data, e.g. stored in cloud repositories. [Ganchev, 2013] proposes the context to include both the data sensed in the environment (as in a typical context-aware system), and the history of the user and the collective history of users who have acted in a similar environment. This constitutes a novel approach in providing context-aware services with elements of community-based personalized information retrieval (PIR), applied to mobile network environments.

4. User-centric ABC&S example

Figure 2 illustrates the user-centric ABC&S concept on the basis of an outgoing call connection service. It shows an example of user A calling user B in the best (i.e. cheapest) possible way. This, of course, will largely depend on the current location of both users (the assumption is that the caller is always aware of the callee's current location). The simplest solution for the caller is to check manually his/her records about the current location of the callee just before calling him/her. For instance, if both users are currently located in their home country (for simplicity it is assumed that the home country is within the EU and is the same for both users), then user A could initiate a call from his/her mobile device through his/her home cellular network to the mobile device of user B, as this is relatively cheap option these days. Alternatively, user A may seek calling user B by utilizing the cheaper option of the Internet telephony, i.e. using the mobile app of some VoIP provider to initiate the call over a Wi-Fi connection, which however could be either paid or free (the latter seems to be the preferred option for low-budget users). Consider now that the caller, user A, moves to another country. Of course using VoIP over Wi-Fi is the cheapest option again. However, within the EU, the cost of international roaming calls is going down all the time (and currently stands as 23.37c/min to make a call and 6.15c/min to receive a call, when roaming). So this could be an attractive option even for low-budget users who want better quality for their calls. Situation however could be quite different if user A is roaming outside the EU, especially if user B is also currently outside the EU and even in the same country as user A (depending on the country, the roaming cost of the call may go up to several euros per minute in both directions!). Spinning (i.e. buying and using local U/SIM cards) represents one possible solution in this case. Users also have the option of using a VoIP over (free) Wi-Fi, however in doing this they still could miss opportunity of using another VoIP service provider, who supplies better rates in the country, where both users are currently roaming in.

Two possible IT solutions for automatic resolution of this situation are described in the next subsections.

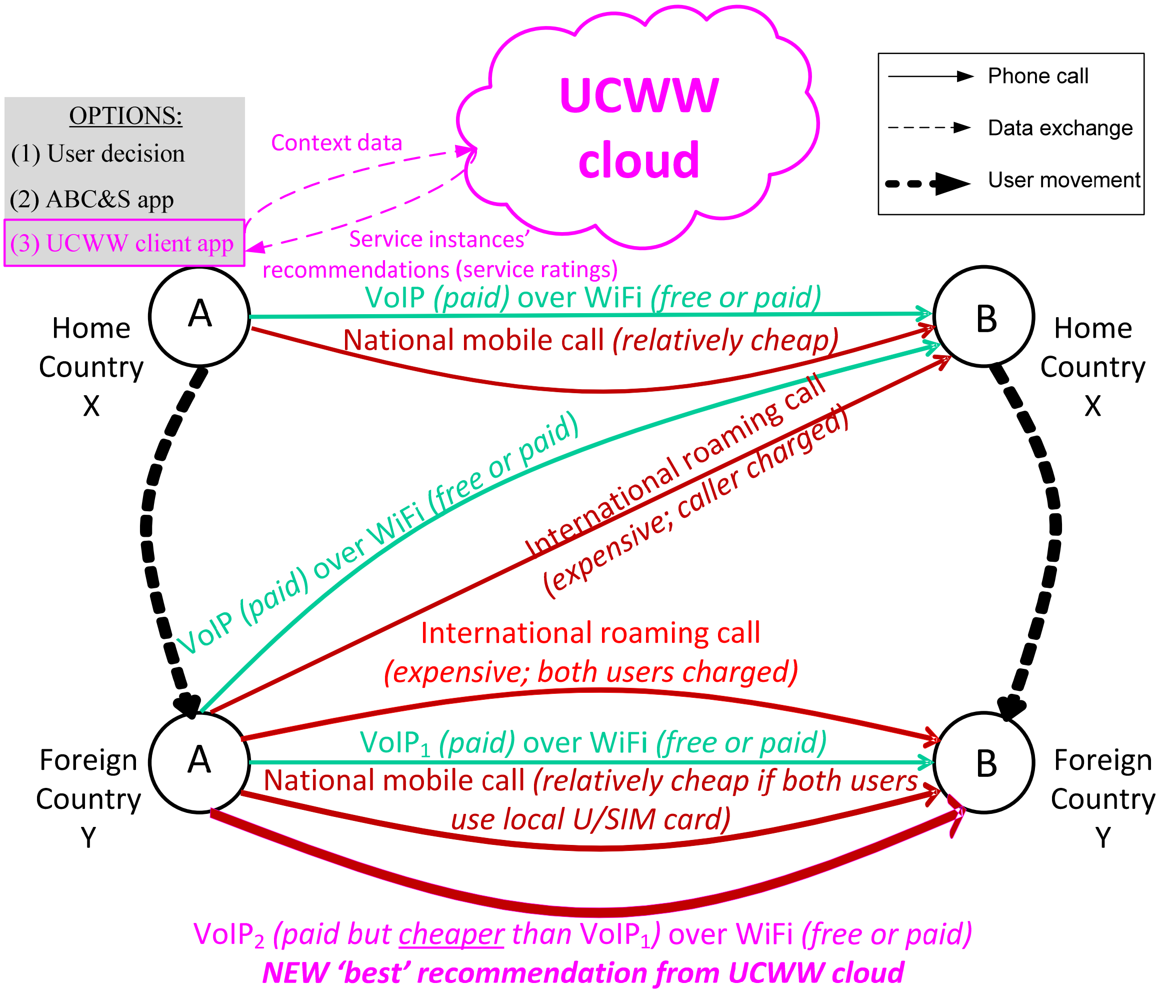

OPTIONS:
(1) User decision
(2) ABC&S app
(3) UCWW client app
Context data
UCWW cloud
Service instances' recommendations (service ratings)
Phone call
Data exchange
User movement
A
Home Country X
B
Home Country X
VoIP (paid) over WiFi (free or paid)
National mobile call (relatively cheap)
VoIP (paid) over WiFi (free or paid)
International roaming call (expensive; caller charged)
International roaming call (expensive; both users charged)
A
Foreign Country Y
B
Foreign Country Y
VoIP₁ (paid) over WiFi (free or paid)
National mobile call (relatively cheap if both users use local U/SIM card)
VoIP₂ (paid but cheaper than VoIP₁) over WiFi (free or paid)
NEW 'best' recommendation from UCWW cloud

Figure 2. A user-centric ABC&S concept illustration

based on an outgoing call connection service example.

Stand-alone ABC&S mobile app solution

The first IT solution one could imagine is to use a generic intelligent ABC&S mobile application as described in [Ji, 2011]. This application sits on the user's mobile device and operates in the background in supporting the user to be fully aware of all options available to call another user (the ABC&S app finds this information by some means, e.g. by searching on the Internet). The app can thus discover relevant service instances and all the necessary details about their offerings sufficient to make informed ABC&S decisions about using them, including knowing how to associate with the access networks to obtain these services in the best possible way. This amounts to a significant advance in consumer-centric ABC&S capabilities and services.

Information about the current location of the callee is also collected and stored by the app. So, whenever user A wants to call user B, s/he just needs to select user B (as a callee) from a list provided by the app. The app then will make the call in the best possible way by exploring all available options, one after another, until the call is successfully made.

The entire process is fully transparent and un-intrusive to the user because the ABC&S decisions are made in the background, following default (e.g., lowest price) or preset ABC&S policies and profile settings. This ‘full transparent ABC&S’ mode means a minimum disruption to the user, and yet it is still consumer-driven ABC&S.

The design and development of such mobile app architecture is described in detail in [Ji, 2011].

Cloud-based solution

The second (and more advanced) IT solution is to use a cloud-based service recommendation system [Ganchev, 2013] as a means for users matching their need to discover the 'best' mobile services, and facilitating, and supporting, the association with them by following a user-driven ABC&S paradigm. A cloud-based UCWW client application [Ganchev, 2015] – associated with such a system – could be used for finding and recommending to users, or even automatically selecting if the user’s profile settings are so set, the ‘best’ mobile services, depending on the current context, including in that decision process the user’s personal profile requirements, e.g., for high-quality voice call service (e.g. 3G/4G) when user A needs to talk with her/his boss, but a Wi-Fi/VoIP lower-quality call service selection (where possible) for talking with friends in order to save money. The complex functional requirements of such client application make for a demanding app design, testing, and validation. A possible design solution, realized through a structured composition of three tiers – a mobile application tier, a web tier, and a cloud tier, is presented in [Ganchev, 2014].

As stated in [Ganchev, 2013], the UCWW cloud (c.f. Figure 2) can operate as a middleware of the context-aware service recommendation system. At the lowest layer, the user’s mobile device collects context data from the environment, and at the highest layer the UCWW client application makes use of this data. Between them operates the middleware of the system, which could be entirely implemented as cloud services. [Ganchev, 2013] describes in detail the flow of context data between a mobile device and the UCWW cloud as well as the mechanism of sending requests and receiving responses from the decision support subsystem, i.e. providing ratings (ranking) of the service providers available for a particular type of service requested by the user.

The main goal here is to design an efficient context-aware middleware for the UCWW cloud by having most of its functions offered as cloud services and the rest running locally on the mobile device. This process requires taking into account a number of aspects [Ganchev, 2013]:

- On the back-end, the UCWW cloud must facilitate the storage of data harvested via mobile devices, and based on the analysis of this data, offer predictions as to the applicability and ABC&S suitability of services to particular users. Over time the data collected relating to particular users can give an accurate view of particular cohorts, based on common interests, repetitive access of particular services, etc. By monitoring this information, the system then can accurately predict the types of services most applicable to individuals, and in turn, recommend these to them.
- Efficient heuristic algorithms must be utilized to facilitate service predictions locally on the mobile devices or as part of the UCWW cloud as an alternative to mining the stored data.
- Within the mobile devices, an effective functional GUI design must be facilitated, with the necessary intelligence to harvest the requisite information to facilitate service predictions. With this in mind, different mobile platforms must be targeted, particularly in the case of the smartphones market, where Android-based devices, iPhones, Windows phones etc. each have a market share.

5. Conclusion

In considering the evolution of a truly Always Best Connected and best Served (ABC&S) enabled wireless communications world, this paper addresses the subjective and objective nature of the ABC&S concept from the viewpoints of the key stakeholders, i.e. the mobile users, access network providers, mobile service providers, and mobile device manufacturers. In particular, it has been shown that the ABC&S definitions and implementations are largely driven by the user requirements, which leads to a user-centric ABC&S realization. This thinking contrasts with supporting ABC&S development through centralized access-network-provider’s management domain as seen by other researchers. A simple example to illustrate this user-centric ABC&S concept, along with corresponding context-related aspects and possible IT solutions, has been provided.

In evolving future wireless world paradigms, the ABC&S concept itself, and the contexts in which it will likely find application, need to be evolved and extended to include some new dimensions, such as adaptive services/applications, open interworking and interoperability of multiple homogeneous and heterogeneous single-access and multi-access wireless networks, reconfigurable devices and network nodes, and operators competition [O'Droma, 2006]. Exploring and defining new ABC&S scenarios, network/service/device environments and techno-business models, implicit in positing ABC&S as a key and integral feature of future generations of wireless communications, necessitates setting down of many new architectural and system design components. These will be the goal of future research.

Acknowledgements

The paper is published with partial support by the project ITHEA XXI of the ITHEA ISS (www.ithea.org) and the ADUIS (www.aduis.com.ua).

The author wishes to specially thank the Director of the Telecommunications Research Centre (TRC) at the University of Limerick (Ireland), Dr. Máirtín O'Droma, for the fruitful discussions and inspiring thoughts on the subject.

Bibliography

[Ganchev, 2007] I. Ganchev, M. O'Droma, S. Poryazov, N. Kalchev. “Consumer-driven ABC&S paradigm realization”. Book of Abstracts of the Jubilee International Conference “New Trends in Mathematics and Informatics” dedicated to 60 years of the Institute of Mathematics and Informatics, Bulgarian Academy of Sciences, Pp. 88-89. 6-8 July 2007, Sofia, Bulgaria. ISBN 978-954-8986-26-7.

[Ganchev, 2013] I. Ganchev, M. O'Droma, N. Nikolov, Z. Ji. “A UCWW Cloud System for Increased Service Contextualization in Future Wireless Networks” (invited paper). Proc. of the 2nd International Conference on Telecommunications and Remote Sensing (ICTRS'13), Pp. 69-78. 11-12 July 2013, Noordwijkerhout, The Netherlands. ISBN: 978-989-8565-57-0.

[Ganchev, 2014] I. Ganchev, Z. Ji, M. O'Droma. “A Cloud-based Service Recommendation System for Use in UCWW”. Proc. of the IEEE 11th International Symposium on Wireless Communication Systems (IEEE ISWCS'2014). Pp. 791-795, 26-29 August 2014, Barcelona, Spain. ISBN: 978-1-4799-5863-4/14. DOI: 10.1109/ISWCS.2014.6933461.

[Ganchev, 2015] I. Ganchev, Z. Ji, M. O'Droma. UCWW Cloud-based ABC&S Mobile App. Proc. of the URSI Atlantic Radio Science Conference 2015 (URSI AT-RASC 2015). 18-22 May 2015, Gran Canaria, Canary Islands.

[Gustafsson, 2003] E. Gustafsson and A. Jonsson. "Always best connected". IEEE Wireless Communications, Vol. 10, No. 1, Pp.49-55, Feb. 2003

[Ji, 2011] Z. Ji, I. Ganchev, M. O'Droma. "An iWBC Consumer Application for 'Always Best Connected and Best Served': Design and Implementation". IEEE Transactions on Consumer Electronics, Vol. 57, No. 2, Pp. 462-470, May 2011, ISSN: 0098-3063. DOI: 10.1109/TCE.2011.5955180.

[McEvoy, 2005] F. McEvoy, I. Ganchev, M. O'Droma. "New Third-Party AAA Architecture and Diameter Application for 4GWW". Proc. of the 16th Annual IEEE International Symposium on Personal Indoor and Mobile Radio Communications (IEEE PIMRC 2005), Vol. 3, Pp. 1984-1988, 11-14 September 2005, Berlin, Germany. ISBN 978-3-8007-2909-8.

[O'Droma, 2003] M. O'Droma, I. Ganchev, G. Morabito, R. Narcisi, N. Passas, S. Paskalis, V. Friderikos, A. S. Jahan, E. Tsontsis, C. F. Bader, J. Rotrou, H. Chaouchi. "Always Best Connected Enabled 4G Wireless World", Proc. of the 12th IST Summit on Mobile and Wireless Communications, Pp. 710-716, 15-18 June 2003. Aveiro, Portugal. ISBN 972-98368-7.

[O'Droma, 2004] M. S. O'Droma and I. Ganchev. "Enabling an Always Best-Connected Defined 4G Wireless World". In: Annual Review of Communications, Vol. 57 (Chicago, Ill.: International Engineering Consortium), Pp. 1157-1170. 2004. ISBN 0-931695-28-8.

[O'Droma, 2006] M. O'Droma, I. Ganchev, H. Chaouchi, H. Aghvami, V. Friderikos. "`Always Best Connected and Served` Vision for a Future Wireless World". Journal of Information Technologies and Control, Year IV, No 3-4, 2006, Pp. 25-37+42. ISSN: 1312-2622.

[O'Droma, 2007] M. O'Droma and I. Ganchev. "Toward a Ubiquitous Consumer Wireless World". IEEE Wireless Communications, Vol. 14, Issue 1, February 2007, Pp. 52-63. ISSN: 1536-1284. DOI: 10.1109/MWC.2007.314551.

[O'Droma, 2010] M. O'Droma and I. Ganchev. "The Creation of a Ubiquitous Consumer Wireless World through Strategic ITU-T Standardization" (invited paper). IEEE Communications Magazine, Vol. 48, Issue 10, October 2010, Pp. 158-165. ISSN: 0163-6804. DOI: 10.1109/MCOM.2010.5594691.

[Tairov, 2011] D. Tairov, I. Ganchev, M. O'Droma. "Third-Party AAA Framework and Signaling in UCWW". Proc. of the 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM 2011), Pp. x1-x5, 23-25 September 2011, Wuhan, China. ISBN 978-1-4244-6252-0, DOI 10.1109/wicom.2011.6040462.

Authors' Information

Ivan Ganchev *– DipEng (summa cum laude), PhD, SMIEEE, ITU-T (Invited Expert), IJTMCC Regional Editor (Europe), CoCoNet'15 Program Chair.*

TRC Deputy Director, University of Limerick, Limerick, Ireland; e-mail: Ivan.Ganchev@ul.ie

Major Fields of Scientific Research: novel telecommunications paradigms, future networks and services, smart ubiquitous networking, context-aware networking, Internet of Things (IoT), Internet of Services (IoS), mobile cloud, trust management, Internet tomography, mHealth and mLearning technologies.

ON AN APPROACH TO STATEMENT OF SQL QUESTIONS IN THE NATURAL LANGUAGE FOR CYBER2 KNOWLEDGE DEMONSTRATION AND ASSESSMENT SYSTEM

Tea Munjishvili, Zurab Munjishvili

Abstract: *The paper focuses on the midterm and final exams in 180 disciplines based on "Cyber1" system developed by the authors. For four years, the exams were held at Sh. Rustaveli State University in the sea-port town of Batumi, Georgia. On the grounds of analyzing the results thereof the authors developed "Cyber2". In the publication they speak about the part the test result analysis plays, its pre-conditions, tasks, means and solutions. They emphasize significance of visualization, generalization and statistical approach to the test results. They also describe method of semantic analysis and algorithms of the questions in the natural language put to a certain section of the database. The programs are in VB. NET.*

Keywords: Analysis of the test results, Semantic analysis

Introduction

Development and use of the knowledge demonstration and assessment systems put forward several tasks, such as making out the optimal timetable of the tests or classes, distribution of disciplines to professors, an automatic understanding of a student's essay, analysis of the test results and the relevant conclusions and recommendations.

As is well known, tests provide important information on the knowledge and demonstration thereof. A test is indispensable for verifying significance of the material imparted to the students, professor's approach to instruction etc.

As a result of analysis of "Cyber1" [Tea Munjishvili, Zurab Munjishvili, 2014] knowledge demonstration and assessment system operated first at Shota Rustaveli State University in the town of Batumi (BSU, Georgia) for four years and then for five years, until 2014/2015 academic year, applied in teaching accounting at the Faculty of Economics and Business at the Tbilisi State University, we developed "Cyber2" [Thea Munjishvili; Zurab Munjishvili , 2015] – the knowledge obtainment, demonstration and assessment software.

Our experience at BSU makes it possible to define conditions **relevant** [Thea Munjishvili; Zurab Munjishvili , 2015] to the functionality and application of “Cyber2” or the other knowledge demonstration and assessment computer systems for that matter:

1. Processing closed and open tests;
2. Setting a task by way of textual or graphic, video or textual and graphic or textual and video information;
3. A prompt to a subject or a test by way of textual or graphic or video information or a combination thereof;
4. Availability of max. three correct answers out of the seven implied ones in a closed test;
5. Giving answers only upon marking the right number of the correct ones and giving “answer” order;
6. Availability of a number answers in open tests;
7. Clicking $q_i \in Q$ mark relevant to $\forall n_i, n_i \in N$ task answer. The mark may be an integer or a decimal positive number;
8. Using words, numbers, sentences or a combination thereof and, also, an abbreviation in an open test;
9. Understanding a statement used in answers in case of desynchronization or insertion of words;
10. In a designational statement and generally answers, writing words in any case and using the wrong variants thereof;
11. Comparison of actual answers to the tasks related to certain subjects, topics or subtopics with the reference value in which case desynchronization, writing words in any case, their insertion or omission will be unacceptable;
12. Formulation the test task by subjects, topics, subtopics and professors;
13. Introducing complex topics or subtopics in the task;
14. Holding examinations by student groups;
15. At the beginning of an examination, arrangement and selection of tests by students in terms of probability;
16. Making out reports after an examination (e.g. a protocol reflecting the course of an examination, examination sheet etc.);
17. Obtainment of analytical information on the results of exams upon completion thereof or at any time;
18. Selection from the database upon putting a question in the natural language;
19. Displaying diagnostic messages, such as using an unknown word, omission of words, numbers etc. during an examination;

20. In case of interruption of an examination for technical etc. reasons, its continuation from the breakpoint;
21. Training: changing the training direction according to the answers given during learning, diagnosing the mistakes and pointing out the ways of prevention thereof.

The necessity of meeting the conditions is detailed in [Thea Munjishvili; Zurab Munjishvili , 2015]. We'll try to prove how important it is to meet the conditions related to obtainment of information on the test result analysis and maintenance of a dialog with the database in the natural language.

Problem Statement

Relevance of reports is unquestionable. For instance: it is necessary to print and familiarize oneself with the test result sheet, test protocol etc. reports, the content of which will have to be predefined.

Apart from that, information search and selection by various parameters is also significant.. Such questions are actually probabilistic and they should be formulated by a management specialist, not a programming professional. The task is related to communication with the databases in the natural language.

Knowledge description methods, such as frames, semantic networks, generating regulations etc. have become widespread in the natural language interactive systems. In most cases a question is put for the purpose of obtainment of the desirable information from the database.

Upon displaying the semantics of a question in the natural language, our task is translation thereof into the one of the database, mainly SQL and then gathering the desirable information.

Along with the uniform algorithmic methods of understanding, the semantics of a question in the natural language, the so-called beyond-the system engineering methods [Thea Munjishvili; Zurab Munjishvili , 2015] relying on the axiom: "***Rules of understanding the semantics can be detected in any problematic area and a logical-semantic model be developed on the grounds thereof***" - are also widespread.

For this purpose, within the knowledge demonstration and assessment "Cyberr2" system, we developed an engineering approach to the semantic analysis of a sentence. It is the so-called "Productive Grammar" method, which is somewhat universal and free from the shortcomings of the tabular suitability principle. The method is discussed in [Z.Munjishvili, 1990], while algorithmic-programming representation of understanding a sentence written in the natural language is detailed in [Thea Munjishvili; Zurab Munjishvili , 2015].

Semantic Analysis Method and Algorithm of Questions to the Database

The objective of our research is understanding a question in the natural language put to the database and generally, realization of semantics of a sentence and its presentation as SQL instruction. We are examining a problematic area, namely, obtainment of the analytical information on the examination results by way of figures and tables. Representing the process as a figure is required by a set of values selected according to a certain criterion(a) marked on X and Y axes.

The content of questions depends on the structure of a certain database but that of any question is typical to SQL instruction. Any question may be considered as a designational sentence made up of more than one word describing searchable items, conditions of selection and classification.

We aim at understanding a sentence at the system entry and translating it into the SQL instruction. There may be omissions or insertion of words or desynchronization in a sentence entered into the system. The words in various cases, synonyms and homonyms may also be used.

The analysis of the structure and content of the questions leads us to the conclusion that in most cases, namely, in the selected problematic area and the task, the logical-semantic model is the basis of presenting examination results as a figure. The model has the following structure: name of the items to be placed on <X axis ; name of those to be placed on a><Y , < selection conditions>.

The selection structure is as follows =:<object><temporal parameters: a semester, an academic year><instruction status><instruction stage>.

Statements are acceptable in terms of questions put for the sake of obtainment the examination results

$$G = \bigcup g_j,\ \mathrm{j} = 1....\mathrm{n}.$$

Let us formulate the G set conditions as follows:

G is a pre-known definite set, with the sentences in the natural language or the order of words being its elements.

1. g_i- marks a sentence with “i” as its conditional number, while $g_{i,\lambda}$ – a word thereof with λ as its conditional number. Then the $g_{i,\lambda}$ used words make up dictionaryL. Composition of G depends on the structure of the system, namely the Cyber 2 database, the knowledge demonstration and assessment by subjects and the appraisal system.
2. Any two elements of G differ at least by a single word;

$$\forall(g_i, g_{i_0} \in G) \Rightarrow (g_{i_0} \setminus (g_{i_0} \cap g_i) \neq \{\} \wedge g_i \setminus (g_{i_0} \cap g_i) \neq \{\})$$

3. Any pair may contain the words similar in content.
4. In Cyber 2 knowledge demonstration and assessment system, a certain selection condition (action) corresponds to a question or a phrase - ψ_i , i.e. according to $g_i \rightarrow \psi_i$. logical – semantic model, the selection condition may contain one or more words. ψ_i may be a word, a sequence thereof, a set of symbols etc. Consequently, the G set $g_i \in G$ elements are reflected in G^* set of operations $\psi_i \quad \in G^*_{,} f : G \rightarrow G^*$, $f(g_i) = \psi_i$.

The desired results are obtained if $\forall g_i \in G$, then g_i may be regarded as the product, the $a_k \in L$ words included in g_i as conditions and the names of objects to be placed on X and Y axes, while ψ_i as selection conditions corresponding to g_2 as operation. In this sense, “the analysis” of a question – statement in the selected problematic area may be brought down to the formation of a productive system, the dictionary arrangement and the SQL instruction relevant to the incoming facts.

As exemplified by Cyber 2, method of the semantic analysis and the algorithm is based on those of understanding the answers to the open tests. The method and algorithms are detailed in [Tea Munjishvili; Zurab Munjishvili , 2015].

In Cyber 2, two tables reflecting the examination results, such as tbmain (information on a student’s activities, the examinations) and the tbmosasmeni (list of the students to do a repeated course of the same subject) form the basis of presenting them by way of a figure and understanding questions

By scores, the tbmain table contains the result –related information detailing a student’s activities from the very beginning of an ‘i’’ course (an activity, a midterm exam, the final, the repeated exam) and the order of appraisals.

The principles presented by way of the productive ones and formulated according to the logical-semantic model of the questions form the algorithmic basis of understanding them In the case in question, the structure of the productiveprinciples is as follows:

Conditions: name of the object to be placed on X axis, name of the object to be placed on Y axis

action: selection condition(s)

In this case, the following principles were formulated:

P1: Subjects, appraisals ⇨ <a student> AND <optional selection conditions >

P2: studens, appraisals,⇨ <subject> AND <<optional selection conditions

P3: subjects, quantity ⇨<appraisal> AND <<<optional selection conditions >

P4:groups, quantity ⇨<appraisal> AND subject AND <optional selection conditions>

In this case, the optional selection conditions are: appraisal type, number, semester, academic year, instruction stage and status.

A dictionary made up of three ones (table) $L = L_1 \cup L_2 \cup L_3$.

L_1 is the structure of an entry

<L_1 a dictionary entry>:=<word $a_k \in g_{i,\lambda}$><$a_k \in g_{i,\lambda}$ a word or its wrong variant or a synonym>

L_1 the words used in the productive principles (textual data) are entered into the dictionary

L_2 the structure of an entry:

<L_2 < a dictionary entry>:=the$a_k \in g_i$ morphological root of the right version of words in the dictionary>< name of the word by the base table>

L_2 dictionary is compiled along with L_1. The system enters an L_1 word unchanged The administrator edits it and creats its morphological root. There are no wrong variants in the dictionary.

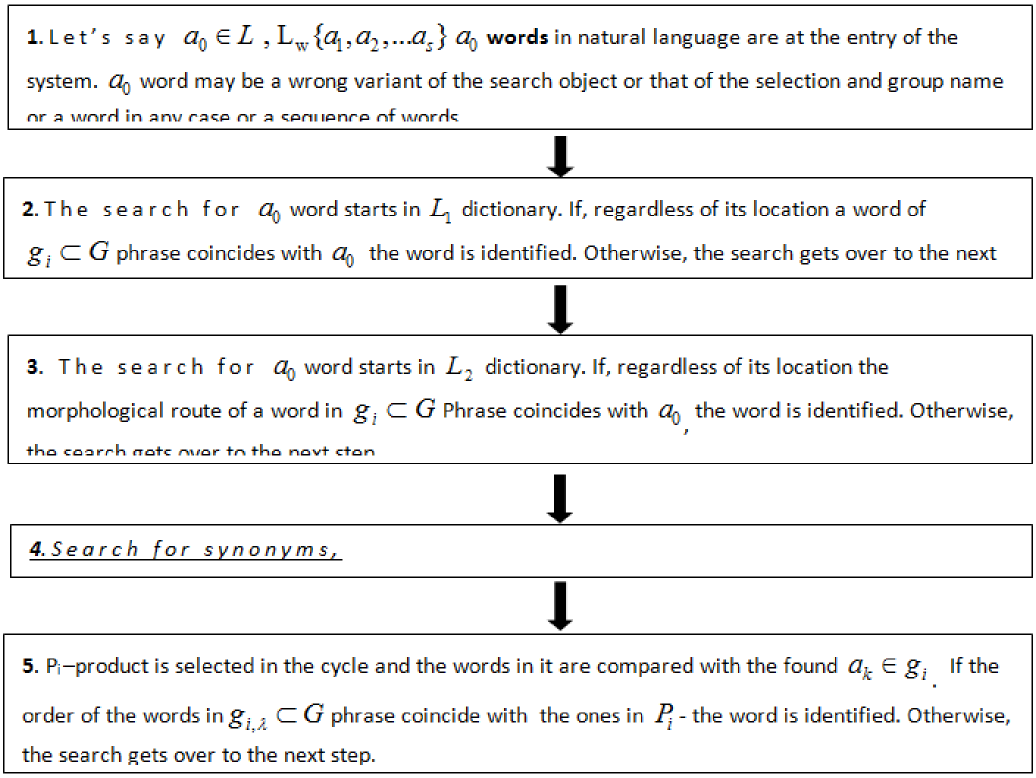
L_3 The dictionary structure is as follows

<L_3 dictionary entry >:=<naming the selection parameter by the base table ><a sentence made up of the $a_k \in g_i$>words in L_1 dictionary.

L_3 dictionary is compiled by the administrator

Understanding a question and its presentation by way of SQL instruction: /fig. 1/

Understandingly, for the discussed problematic area, there is no need to formulate a question in the natural language and apply the specified complex pattern in order to understand its semantics. In the case in question, the task may be solved by a simple selection from the lists or another method. The objective of the article is to highlight “a forgotten” problem: a dialog with databases in the natural language.



1. Let's say $a_0 \in L$, $L_w\{a_1,a_2,...a_s\}$ a_0 **words** in natural language are at the entry of the system. a_0 word may be a wrong variant of the search object or that of the selection and group name or a word in any case or a sequence of words

2. The search for a_0 word starts in L_1 dictionary. If, regardless of its location a word of $g_i \subset G$ phrase coincides with a_0 the word is identified. Otherwise, the search gets over to the next

3. The search for a_0 word starts in L_2 dictionary. If, regardless of its location the morphological route of a word in $g_i \subset G$ Phrase coincides with a_0, the word is identified. Otherwise, the search gets over to the next step

***4.** Search for synonyms,*

5. P_i–product is selected in the cycle and the words in it are compared with the found $a_k \in g_i$. If the order of the words in $g_{i,\lambda} \subset G$ phrase coincide with the ones in P_i - the word is identified. Otherwise, the search gets over to the next step.

Fig. 1. Understanding and Presenting a Question by Way of SQL Instruction.

Conclusion

1. By means of generalization of theoretical issues and on the grounds of handson experience, we developed the conditions of functionality and application of the knowledge demonstration and appraisal computer systems, namely obtainment of the analhtical information on the test results by way of diagrams, to this end, formulation of questions in the natural language and understanding a question in case of desynchronization, insertion of words, putting them into any case or using a wrong variant of a word;
2. A question is understood by means of the logical-semantic model and application of the knowledge demonstration method by means of productive principles widespread in the artificial intellect;
3. After further research, the described approach to understanding questions to the database presented in the diagrams that reflect the analytical data regarding the examination results may be applied to the other problematic areas, as well.

Bibliography

[Tea Munjishvili, Zurab Munjishvili, 2014] Tea Munjishvili, Zurab Munjishvili. Knowledge Demonstration and Assessment System “Cyber1”, international Journal "Information Technologies & Knowledge" Volume 8, Number 3, 2014, pp. 271-279.

[Munjishvil T., Munjishvil Z., Nakashidze V, 2014] Munjishvil T., Munjishvil Z., Nakashidze V. System of knowledge revealing and rating – “Cyber 2”. 9th MIBES ANNUAL INTERNATIONAL CONFERENCE 2014 THESSALONIKI, GREECE, 30/5– 1/6 .CD ISBN# 978-960-93-6161-3. pp. 111-121.

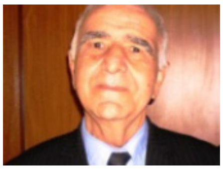
[Tea Munjishvili; Zurab Munjishvili , 2015] Tea Munjishvili; Zurab Munjishvili. "The system of Discovery and Estimation of Knowledge “Cyber2”", Scholars' Press, Saarbrücken HRB 18918. Published on: 2015-01-15 Number of pages: 108. Book language: English. ISBN-13: 978-3-639-76094-1.

[Z.Munjishvili, 1990] Z.Munjishvili. Problem-oriented method of semantic analysis for sentence of natural language. Collection of Knowledge, dialogue, decision, Kiev, Ukraine, “Naukova Dumka”, 1990.

Authors' Information

Tea Munjishvili *– Iv.Javakhishvili Tbilisi State University, 0129-Tbilisi, Georgia. Associated Professor, e-mail: tmunjishvili@gmail.com*

Major Fields of Scientific Research: Practical information systems research, Logical-probability models while assessing risk by enterprises.



Zurab Munjishvili *– Interbusiness Acadeny, 8, Shalva Djaparidze str.*

Tbilisi, 0178, Georgia. Associated Professor, e-mail: zurab_ztm@ rambler.ru

Major Fields of Scientific Research: General Practical information systems research, Creator for knowledge E-Systems.

NEAREST NEIGHBOR SEARCH AND SOME APPLICATIONS

Hayk Danoyan

Abstract: *The problem of finding of nearest neighbors from the data set for a given query is considered. The important applications for different data types when the NN algorithms may be applied are enumerated namely for Textual data, Image data and Genome data. For each individual case the models under consideration (data representation type, similarity measure etc.) are those, which keep in view the specifications from practice.*

Keywords: *Nearest neighbor, k-NN, text classification, parameter estimation, human pose estimation, sequence alignment, homology search, protein secondary function prediction*

ACM Classification Keywords: *H.3.3, I.2.10, I.7.*

1. Introduction

Let we have some set V and some distance measure $\rho(u,v)$ between two elements u,v of V. For arbitrary subset A and vector x by $\rho(x,A)$ we denote the following $\rho(x,A) = min_{a\in A}\, \rho(x,a)$. The subset $F \subseteq V$ and a vector $x \in V$ are given. Consider the problem of finding the nearest elements (by means of distance $\rho(u,v)$) of F from x named as NN (nearest neighbor). In other words it is required to find the set$\mathcal{S} = \{y \in F/\rho(x,y) = \rho(x,F)\}$. The other variation of the problem also often considered named as “k-nearest neighbors” or short “k-NN” problem which consist in finding a set $\mathcal{S}$ such that:

I. $|\mathcal{S}| = k$;

II. $\forall y \in \mathcal{S}$ and $\forall z \in V\backslash\mathcal{S}$, $\rho(x,y) \le \rho(x,z)$.

Keeping in mind the big volume of F which may considered in applications the main requirement is to exclude the linear scan which can made the problem practically unsolvable. The approximate variant of the problem is also considered named as approximate nearest neighbor problem [Andoni, 2009; Gionis, 1999].

There are a large amount of literature devoted to case when $V = \{0,1\}^n$, and $\forall x, y \in V$ by $\rho(x,y)$ is denoted the Hamming metric for example [Gionis, 1999; Rivest, 1974; Aslanyan, 2013; Aslanyan, Danoyan, 2013; Aslanyan, 2014], etc. The R^n where R is the set of reals under metric of ρ_p is also under consideration, where $\rho_p(x,y) = \sqrt[p]{\sum_{i=1}^{n} |x_i^p - y_i^p|}$ for any $x, y \in R^n$. The aim of this paper is to bring survey on known applications where nearest neighbor method may be applied.

2. Orthography Correction with Dictionary

Let we have a dictionary of any language. Someone insert a word which may, in fact, contain an orthographic error. The aim is to detect the error. There are types of errors which can occur more likely: namely one can miss some character, type other character instead of right character or type an excessive character. Here we demonstrate the mentioned types of errors on word “character”:

- caracter (missed the letter “h” after “c”);
- charakter (instead of letter “k” mast be “c”);
- charactere (letter “e” at the end is excessive).

The technic is the following: as a set F mentioned above we keep the set of orthographic forms of words of the language. The distance between words we consider the edit distance [Wagner, 1974]. So for the typed word we find the nearest neighbors from F by means of edit distance, which can allow correcting the error in the typed word.

3. Text Categorization

Let we have a set of documents $D = \{d_1, \dots, d_n\}$ and set of categories $C = \{c_1, \dots, c_k\}$. For each document $D_j, j = 1, \dots, n$ we have a label denoted as $y_j \in C$.

The problem is for unknown test instance d to predict the value of category vector [Sebastiani, 2002] or more formally it is required to approximate the unknown function $\Phi: D \to C$ where

$$\Phi(d) = c_i \Leftrightarrow \text{document } d \text{ belongs to class } c_i \text{ for some } i \in \{1, \dots, k\}. \quad (3.1)$$

The more general case when the classes may overlap can be considered [Sebastiani, 2002]. The text categorization problem is naturally arises in many applications. We mention the following [Sebastiani, 2002], [Aggarwal, 2012]:

Document Filtering. Let we have a collection of news articles. As the number of electronic articles written per day may be very big so it will become necessary to create a system which will automatically label each article to corresponding class (for example politics, art, science, business, sport, technics etc).

Document organization and retrieval. Let one has an electronic collection of text documents (scientific articles, news articles, web collections, etc.). It is required to organize document hierarchically such, that the browsing and retrieval will be efficient.

Spam Detection. When one receive the electronic email it may be spam. So in order to prevent the loss of user's time for reading it one need to categorize it spams or not spam.

For other possible applications we will refer to [Sebastiani, 2002; Aggarwal, 2012].

3.1. Text Representation

At first we have to represent the text document (which may be for example in format pdf, docx, etc.). The one of possible approaches is "bag-of-words" [Sebastiani, 2002] when each document is represented as $d_j = (w_{j1}, \dots, w_{j|T|})$, where T is a set of all possible terms, and w_{ji} is the weight of i^th^ term in j^th^ document. The set of all words occurring at least in one of the documents can be considered as T [Aggarwal, 2012, Salton, 1988, Sebastiani, 2002]. The articles (a/an/the etc), prepositions, conjunctions etc. and topic-neutral words are ignored. The second stage is stemming [Sebastiani, 2002] (grouping the words having the same morphological root).

There are different ways of defining weight [Sebastiani, 2002; Aggarwal, 2012; Salton, 1988]. One of the simplest ways is the following (also called the set-of-words representation): $w_{ji} = 1$ if i^th^ term appear in j^th^ document and $w_{ji} = 0$ otherwise. In general weights belong to range [0,1].

Probably one of the most used weighting method is:

$$w\big(d_i, t_j\big) = \phi(d_i, t_j) \log \frac{n}{v(t_j)}, \tag{3.1.1}$$

where $\phi(d_i, t_j)$ denotes the number of times t_j occurs in d_i, and $v(t_j)$ denotes the document frequency of term t_j, that is, the number of documents in Tr in which t_j occurs.

In order for the weights to fall in the [0,1] interval and for the documents to be represented by vectors of equal length, the weights resulting from $w\left(d_i, t_j\right)$ are often normalized by cosine normalization, given by the following formula

$$w_{ij} = \frac{w(d_i,t_j)}{\sqrt{\sum_{k=1}^{|T|} w_{ik}}}. \tag{3.1.2}$$

Although normalized $w\left(d_i, t_j\right)$ is the most popular one, other indexing functions have also been used, including probabilistic techniques [Sebastiani, 2002; Salton, 1988].

The function representing the similarity of two document representing vectors may be useful, which may be defined as:

$$\rho\left(d_i, d_j\right) = \sum_{k=1}^{|T|} w_{ik} w_{jk}. \tag{3.1.3}$$

Mention that in case of binary text representation function defined by (3.1.3) coincides with hamming metric.

3.2. Text Classification Using NN Method

The dimensionality reduction technics may be applied to decrease the dimensionality of data type. For a survey for such technics we refer to [Sebastiani, 2002; Yang, 1997].

The nearest neighbor methods may be applied by the following way: Let the set of nearest neighbors of document d be the $(d_{i_1}, \dots, d_{i_k})$ and$(y_{i_1}, \dots, y_{i_p})$ are the corresponding labels. Now let $B_i = \{d \in F_x / \Phi(d) = c_i\}$. The label of document d to be classified will be assigned a label y where y defined as

$$y = \underset{i}{\operatorname{argmax}} |B_i|. \tag{3.2.1}$$

As we already mentioned that this approach may easily be generalized for case with overlapping classes, i.e. the sample may belong to more than one class.

4. Human Pose Estimation

The articulated pose estimation problem is formulated as follows [Shakhnarovich, 2005; Shakhnarovich, 2003]. We are given an image which contains a human body. We also have an articulation model – a model of the body that describes the current 3D body configuration in terms of a set of limbs and rotational joints that connect them into a tree structure.

One can synthetically generate image of a humanoid with a computer graphics program like Poser (Figure 4.1). That image will correspond to the articulated model. The model is shown by plotting 2D projections a number of key joints (crosses) and the lines connecting them, which roughly correspond to limbs Figure 4.2. This model may be described by set of numbers, namely the coordinates of the joints. In fact, there are hundreds of parameters in these numbers that affect the resulting image: the articulated pose of additional body parts not accounted for by this coarse model, such as fingers; shape of the actual body parts, facial expression etc. Added to that could be the parameters that describe the scene, the objects in the background etc.

The goal of a computer graphics program like Poser is to start with these parameters and produce a realistic image.

The goal of computer vision is the opposite. In the context of articulated pose estimation this goal is to start from the Figure 4.1, and recover the relevant parameters (Figure 4.2) of the representation that "generated" the image, while ignoring the nuisance parameters.

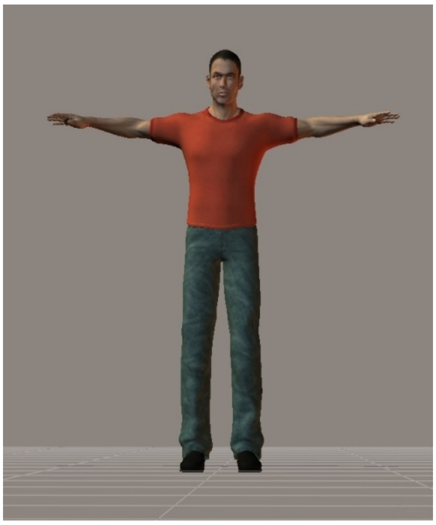


Figure 4.1

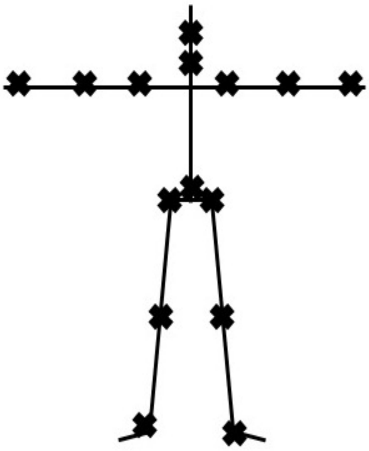


Figure 4.2

4.1. Example Based Pose Estimation

Now let we have a human image I represented as vector x (x belongs to some vector space X) and we want to retrieve the corresponding parameter vector denoted by θ belonging to some metric space Θ [Torralba, 2008]. The distance between two vectors $q, t \in \Theta$ will be denoted by $\rho_\Theta(q, t)$. We also have for some image feature vectors $x_1, x_2, \cdots x_m$ their corresponding parameter vectors i.e. $\theta_1, \theta_2, \cdots, \theta_m$. In example in the previous section parameter vector corresponding to image are the angles between those parts of body which are connected by joints (Figure 4.2). One approach to estimate θ based on k-NN is [Cover, 1968]

$$\theta = \frac{1}{k} \sum_{x_i \in F_{x,k}} \theta_i. \tag{4.1.1}$$

For theoretical ground of formula (4.1.1) we refer to [Cover, 1968].

In formula (1) the unknown parameter is estimated by using parameters corresponding to k nearest neighbors of x by means of metric ρ_X. Let us mention the following two difficulties in connection to formula (4.1.1):

- When the dimension of space X is big which is usual for multimedia applications the problem of searching k-NN may become to linear scan which is unacceptable in practice [Gionis, 1999]. Mention that in [Gionis, 1999] proposed method for approximate nearest neighbor search in sublinear time.
- The parameter values corresponding to k nearest neighbors of x may not be “close” to θ by means of metric ρ_Θ (Figure 4.1.1).

To come over the mentioned problem it is proposed [Shakhnarovich, 2005; Shakhnarovich, 2003; Torralba, 2008; Cipolla, 2013; Shakhnarovich, Darrel, Indyk, 2005] to consider the hash schema $H = \{h_1, \dots, h_n\}$, $H: X \to E^n$ and $h_i: X \to \{0; 1\}$ such that the length of the hash code will be relatively small (to perform effective nearest neighbor search) and the “close” points in Θ by means of metric ρ_Θ will have “close” hash values in corresponding Hamming space with high probability.

We will treat the similarity as some binary relationship $S: X^2 \to \{-1, +1\}$ [Shakhnarovich, 2005], i.e. we will suppose two human images I_1 and I_2 are similar if $S(x_1, x_2) = 1$ and are not similar if $S(x_1, x_2) = -1$. In [Shakhnarovich, 2005] it is considered the case when

$$S(x_1, x_2) = \begin{cases} +1 \ if \ \rho_\Theta(\theta_1, \theta_2) \le R, \\ -1 \ if \ \rho_\Theta(\theta_1, \theta_2) > R, \end{cases}$$

for some predefined value of R, where x_1 and x_2 are the corresponding feature vectors of I_1 and I_2. The meaningful interpretation of this formula in context of pose estimation problem is the following: two images are similar if they have “close” poses or “close” values of joint angel vectors.

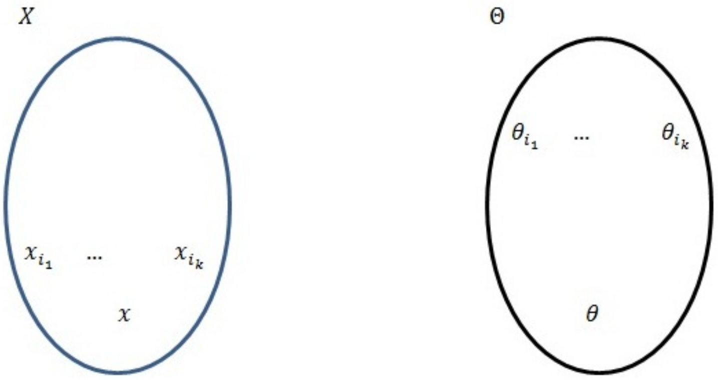


Figure 4.1.1

The general structure of hash function is [Shakhnarovich, 2005; Shakhnarovich, 2003; Torralba, 2008]:

$$H(x) = (h_1(x), \dots, h_n(x)), \tag{4.1.2}$$

where $\alpha_1, \dots, \alpha_n$ are real numbers.

Each $h_i \ i = 1, \dots, n$ has the form [Shakhnarovich, 2005; Shakhnarovich, 2003]:

$$h_{T,f}(x) = \begin{cases} 1 \ if \ f_i(x) \le T_i, \\ 0 \ if \ f_i(x) > T_i, \end{cases} \tag{4.1.3}$$

where $f_i\colon X \to R$ and T_i is some threshold. As f one can usually use the projection function [Shakhnarovich, 2005; Shakhnarovich, 2003; Gionis, 1999], i.e. $f(x) = x_d$, where by x_d is denoted the d^th component of the vector x.

With each function h we can define a classifier by the following way

$$c(x, y) = \begin{cases} +1, if \ h(x) = h(y) \\ -1, otherwise. \end{cases}$$

As a measure of classification accuracy we will consider the following two values: expected true positive rate

$$R_{tp} = \mathop{\mathrm{E}}_{\substack{(x,y) \\ S(x,y)=+1}} [\Pr(c(x, y) = +1)]$$

and expected false positive rate

$$R_{tn} = \underset{\substack{(x,y)\\ S(x,y)=-1}}{\mathrm{E}} [\Pr\,(c(x,y) = +1)].$$

These values will be estimated from the available examples of similar and dissimilar pairs [Shakhnarovich, 2005; Shakhnarovich, 2003]. The empiric true positive rate will be denoted as

$$\hat{R}_{tp} = |\{(x,y)/S(x,y) = +1, c(x,y) = +1\}|.$$

By the same manner we will denote the false positive empirical rate:

$$\hat{R}_{fp} = |\{(x,y)/S(x,y) = -1, c(x,y) = 1\}|$$

The algorithm computing the optimal value of the threshold is brought in [Shakhnarovich, 2005].

Below the Algorithm 4.1.1 which computes the value of n and constructs the hash-function $H: X \to E^n$ is brought:

Algorithm 4.1.1: Similarity Sensitive Coding

```
Given: Set of similarity-labeled pairs $((x_1, y_1), l_1), \dots, ((x_N, y_N), l_N)$,
Given: Lower bound on $\hat{R}_{tp} - \hat{R}_{fp}$ gap $G$
Output: Embedding $H: X \to E^n$ (n to be determined by the algorithm).
Let $n := 0$.
Assign equal weights $W(i) = 1/N$ to all N pairs.
for all d = 1, . . . , dim(X) do
    Let $f(x) = x_d$
    Apply Algorithm of finding best threshold to obtain a set of m thresholds $\{T_t^d\}_{t=1}^m$ and associated true positive $\{\hat{R}_{tp_t}^d\}_{t=1}^m$ and false positive $\{\hat{R}_{fp_t}^d\}_{t=1}^m$ rates
    for all $t = 1, \dots, n$ do
        if $\hat{R}_{tp_t}^d - \hat{R}_{fp_t}^d \geq G$ then
            $n := n + 1$;
            $h_n(x) = h_{f,T_t^d}(x)$;
end for
Return H = $[h_1, \dots, h_N]$
```

5. Genomic Sequence Processing

The nearest neighbor approach has also many biological applications. Let we have a finite alphabet Σ (for biological applications it is useful to consider when $|\Sigma| = 4$ and $|\Sigma| = 20$ corresponding to the cases of nucleotide sequences and amino-acid sequences). Let us consider two strings $a = a_1 a_2 \dots a_n$ and $b = b_1 b_2 \dots b_m$, where $a_i \in \Sigma, b_j \in \Sigma,\ 1 \le i \le n,\ 1 \le j \le m.$ To refer the ith element of string a we will use the notation $a[j]$. For $i < j$ we use $a[i, j]$ as an alternative notation of string $a[i] \dots a[j]$. Consider we have an operation of insertion the symbol "-" somewhere in string a or b.

An alignment (global) [Gusfield, 1997; Mount, 2013] A of two strings a and b is a finite set of transformations by inserting the symbol "-" in a or b, such that after these transformations we get correspondingly the sequences a' and b' of the same length l.

We escape the case when $a'[i] = b'[i] = "-"$ for some i, $1 \le i \le l$. Let us denote the set of all possible alignments by $\mathcal{A}$. It is obvious that $\mathcal{A}$ is finite. For each alignment A we define its value

$$S_A(a,b) = \sum^{l} \underset{a'[i]\neq\text{"-" or } b'[i]\neq\text{"-"}}{{}_{i=1}} s(a'[i], b'[i]) - kw, \tag{5.1}$$

where s is a some function defined over $\Sigma \times \Sigma$, k is the number of symbols "-" in alignment and w is a number which can be interpreted as a "cost" of each occurrence of symbol "-". The function s can be represented as a matrix of size $|\Sigma| \times |\Sigma|$. Such matrixes are called substitution matrixes. For detailed information about calculation of these matrixes we refer to [Dayhoff, 1978; Henikoff, 1992].

The optimal alignment is an alignment A_o such that has maximal value which we will denote by $S(a, b)$.

$$S(a,b) = \max_{A \in \mathcal{A}} S_A(a,b). \tag{5.2}$$

The local alignment is alignment of substring $a[i, j]$ and $b[p, q]$ such that have the maximal value of global alignments overall substrings of a and b, i.e.

$$S_{loc}(a,b) = \max_{\substack{0 \le i \le j \le n \\ 0 \le p \le q \le m}} S(a[i,j], b[p,q]). \tag{5.3}$$

We call the number $S_{loc}(a, b)$ the value of local alignment. Algorithms computing local and global alignments for given two sequences and substitution matrix were brought in [Needleman, 1970; Smith, 1981].

Now let we have a set of strings maybe of different lengths. For the given sequence it is required to find all sequences most "similar" to the given sequence by means defined above (the score of local or global alignment is maximal). The mentioned problem named as sequence nearest neighbor problem. There may be different variations of notion “similarity” namely gaps may be allowed with different cost functions [Waterman, 1976], or the edit distance and some other distances may be considered. The appropriate selection of similarity measure depends on concrete application.

6.1 Search for Homologous Proteins

At the present there are available many protein databases [Altschul, 1990]. For the new sequenced protein/gene etc. the problem rises to find its properties/functions [Mount, 2013; Gabaldon, 2004; Sleator, 2010]. The experimental way may be costly by means of time/money etc. One of the possible approaches based on homologous proteins with known functions. Two sequences will be called homologous if they have the same ancestor. Proteins having similar sequences are usually homologous. So the function of unknown protein may be predicted by knowing the function of homologous (similar) proteins.

There exist empiric algorithms finding similar sequences for a given sequence (Blast, Fasta etc.) [Mount, 2013; Gusfield, 1997; Altschul, 1990]. Here the meaningful implementation of the algorithm BLAST for proteins [Mount, 2013] is brought below:

Input: Query sequence S[1,n]

- For each sequence a list of words of length 3 L={S[1,3], S[2,4],…,S[n-2,n]} using substitution matrix Blossum62 construct all words with alignment score $\geq$some threshold T. Such words will be called hits;
- For each hit scan the database for exact-match;
- Extend the alignment.

Return high-scoring alignments.

Mention that BLAST is heuristic algorithm. For analyzing of implementation we refer to [Altschul, 1990].

6.2 Protein Secondary Structure Prediction

Now let we have a protein database and for each protein the corresponding secondary structure is available in database. For the protein with unknown secondary structure it is required to predict its secondary structure using the database. The nearest neighbor method may be applied also, to keeping in mind the hypothesis that the homologous sequences have the same secondary structure tendencies [Levin, 1986]. The algorithm described in [Levin, 1986; Levin, 1997] and [Keedwell, 2005]

Algorithm 6.2.1

```
Input Sequence S of length n, integer m, threshold q.
For each k=1, k≤n-m+1, k++
find set $F_{S,k}$ of sequences of length m from database which align with S[k,k+m] with score ≥q.
Consider the matrix a n×p comment: p-the number of secondary structure conformation, usually p=3, [Keedwell, 2005].
In the ith row and jth column of the matrix write sum of all alignment scores in $F_{S,k}$where in the i-th place corresponds j-th secondary structure conformation.
Return $\alpha_1, \dots, \alpha_n$ where $\alpha_i = \max_{j=1,\dots,p} a_{ij}$.
```

For example of the performance of algorithm we refer to [Keedwell, 2005].

Conclusion

The problem of nearest neighbor search becomes important in many applied domains such as:

- Textual data mining (document filtering, spam detection, plagiarism detection, etc.),
- Machine vision (human pose estimation, image classification, image search, etc.),
- Computational biology (Search for homologous proteins, protein secondary structure prediction, etc.).

The major difficulty is the amount of available data, when the linear scan becomes practically unrealizable. So the requirements on algorithm are to consider data points from the database as few as possible and to be effective from computational viewpoint.

Bibliography

[Aggarwal, 2012] C. Aggarwal, C. Zhai, Mining text data, Springer, 522pages, 2012

[Altschul, 1990] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman, Basic local alignment search tool, Journal of Molecular Biology, 215, pp. 403-410, 1990

[Andoni, 2009] A. Andoni, Neares Neighbor Search: the Old the New, and the Impossible, PhD. Thesis, MIT 2009, 178p.

[Aslanyan, 2013] L. H. Aslanyan, H. E. Danoyan, On the Optimality of the Hash-Coding Type Nearest Neighbour Search Algorithm, Selected works of 9th CSIT conference, pp. 1-6, 2013

[Aslanyan, 2014] L. H. Aslanyan, H. E. Danoyan, Complexity of Hash-Coding Type Search Algorithms with Perfect Codes, JNIT Journalof Next Generation Information Technology, Vol. 5, number 4, pp.26-35, 2014

[Aslanyan, Danoyan, 2013] L. H. Aslanyan, H. E. Danoyan, On the optimality of a hash-coding type search algorithm, Proceedings of the 9th conference CSIT, Yerevan, Armenia, pp. 55-57, 2013

[Cipolla, 2013] R. Cipolla, S. Battiato, G. M. Farinella, Machine learning for computer vision, Springer 2013

[Cover, 1968] T. M. Cover. Estimation by the nearest neighbor rule. IEEE Transactions on Information Theory, 14:21–27, January 1968,

[Dayhoff, 1978] M. Dayhoff , R. Schwartz, B. Orcutt, A model of evolutionary change in proteins, Atlas of protein sequence and structure, Vol. 5, No. suppl 3. pp. 345-351, 1978,

[Gabaldon, 2004] T. Gabaldon, T; M. A. Huynen, Prediction of protein function and pathways in the genome era". Cellular and Molecular Life Sciences 61 (7-8): pp. 930–944, 2004

[Gionis, 1999] A. Gionis, P. Indyk, R. Motwani, Similarity Search in High Dimensions via Hashing, Proceedings of the 25th VLDB Conference, Edinburg, Schotland, pp.518-529, 1999

[Gusfield, 1997] D. Gusfield, Algorithms on Strings, Trees and Sequences, Cambridge University Press, 534 pages, 1997

[Henikoff, 1992] S. Henikoff and J. Henikoff, Amino acid substitution matrices from protein blocks, Proc. Natl. Acad. Sci USA., vol. 89, pp.10915–10919, 1992

[Keedwell, 2005] E. Keedwell and A. Narayanan, Intelligent Bioinformatics, Wiley, 280p. 2005

[Levin, 1986] J. M. Levin, B. Robson and Jean Gamier, An algorithm for secondary structure determination in proteins based on sequence similarity, FEBS, vol. 205, num. 2, pp. 303-308, 1986

[Levin, 1997] J. M.Levin, Exploring the limits of nearest neighbor secondary structure prediction, Protein Engineering vol.10 no.7 pp.771–776, 1997

[Mount, 2013] D. Mount, Bioinformatics: Sequence and Genome Analysis, second edition, Cold Spring Harbor Laboratory Press, 665 pages, 2013,

[Needleman, 1970] S. Needleman, C. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, J Mol Biol.; 48(3):443-453, 1970

[Rivest, 1974] R. Rivest. On the optimality of Elias's algorithm for performing best-match searches. Information Processing, pp. 678–681, 1974

[Salton, 1988] G. Salton and C. Buckley, Term-Weighting approaches in automatic text retrieval, Information Processing & Management Vol. 24, no. 5, pp. 513-523, 1988

[Sebastiani, 2002] F. Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys, 34(1), 1-47pp., 2002,

[Shakhnarovich, 2003] G. Shakhnarovich, P. Viola, and T. Darrell. Fast pose estimation with parameter-sensitive hashing. In Proc. IEEE International Conference on Computer Vision, volume 2, 2003,

[Shakhnarovich, 2005] G. Shakhnarovich, Learning Task-Specific Similarity, PhD Thesis, MIT 2005,

[Shakhnarovich, Darell, Indyk 2005] G. Shakhnarovich, T. Darell and P. Indyk, Nearest neighbor methods in Learning and Vision: Theory and Practice, MIT Press, 2005

[Sleator, 2010] R. D. Sleator, P. Walsh, An overview of in silico protein function prediction, Arch Microbiol, no. 192, pp. 151-155, 2010

[Smith, 1981] T. Smith, and M. Waterman, Identifcation of common molecular subsequences, Journal of Molecular Biology, 147, pp. 195-197, 1981

[Torralba, 2008] A. Torralba, R. Fergus, and Y. Weiss. Small codes and large image databases for recognition. In Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2008,

[Wagner, 1974] R. Wagner, M. Fischer, The String-to-String Correction Problem, Journal of the ACM, Volume 21 Issue 1, pp. 168-173, Jan. 1974

[Waterman, 1976] M. S. Waterman, T. F. Smith and W. A. Beyer Some Biological Sequence Metrics, Advances in Mathematics, 20, pp. 367-387, 1976

[Yang, 1997] Y. Yang and J. Pedersen, A comparative study on feature selection in text categorization, In Proceedings of ICML-97, 14th International Conference on Machine Learning, 412–420pp., 1997

Authors' Information

Hayk Danoyan *– Institute for informatics and automation problems of NAS RA, 1, P. Sevak street, Yerevan 0014, Armenia, e-mail: hed@ipia.sci.am*

Major Fields of Scientific Research: Nearest Neighbor Search, Discrete optimization, Coding theory, Machine Learning, Bioinformatics

MOBILE BANKING SECURITY PRACTICES FOR ANDROID USERS

Bonimir Penchev

Abstract: *The widespread usage of mobile phones and smartphones in particular is related with the continuous expansion of their functionalities. Mobile banking is an option which gives users the possibility to perform various banking operations (such as account balance inquiry, account transactions, payments and other basic services available daily in banks) through a mobile device such as mobile phone, smartphone or tablet. A key factor for its wider usage is to increase the level of security, which in turn will increase user's confidence. In this paper, we investigate which of the security features for a safer mobile banking are included in 14 mobile antivirus and security applications for Android OS. In the final assessment are included three additional factors – license, usability and battery usage. On the other hand, based on the main channels for mobile banking (SMS, mobile websites and mobile applications) and the fact that the web browser can be a major source of malware applications, spoofing websites and targeted Trojans, we present how the existence of certain security indicators in mobile web browsers and their recognition would help the user to avoid security attacks. According to the results of our study, we can evaluate and select the combination mobile antivirus and security application - mobile web browser in order to increase user's confidence in mobile banking.*

Keywords: *Mobile security, Mobile banking, Mobile Antivirus and Security Applications, Mobile Web Browser.*

ACM Classification Keywords: *K.4.4 Electronic Commerce – Security*

Introduction

Mobile phones are an integral part of our daily lives. Their usage is not limited to operations like phone calls or text messaging. Technology progressed to such an extent that smartphones' functionalities can be compared to those of contemporary computers. In Smartphone's financial institutions explore an excellent opportunity to provide new type of services. Mobile banking is this type of service that allows the user to carry out various banking operations (such as account balance inquiry, account transactions, payments and other basic services available daily in banks) through a mobile device such as a mobile phone, smartphone or tablet.

Among the benefits that users can derive from mobile banking, there are also different restrictions. The most important of them is related to the security level of this service and it is one of the main user's concerns when deciding whether or not to use mobile banking. In this aspect it is necessary to examine the main critical points in the mobile banking process – bank system, transmission media, mobile device, and user. In our study we will focus on the mobile device.

According to a research conducted by Gartner [Gartner Inc., 2014] for 2013 the market share of smartphones reached 53.6% of overall sales of mobile phones. The same study states that there is an increase of 42.3% compared to 2012. Both these facts lead us to the conclusion that the proportion of the users currently using smartphones is constantly increasing. This is a good enough reason for us to focus our research on this type of mobile devices.

The main smartphone's security threats associated with mobile banking can be the following: banking malware, targeted Trojans, mobile spyware, mobile phishing, smishing, device lost and theft.

Unfortunately, Android OS keeps its leading position as well in terms of its usage – 66.4% [Gartner Inc., 2014], as in the number of developed malware applications – 97% of total mobile malware. [McAfee Company, 2013] These two facts are focusing us on researching mobile antivirus and security applications developed for Android platform.

There are different solutions for dealing with the security problems of smartphones in terms of mobile banking [AV-Comparatives Organization, 2014]:

- Anti-malware – on demand scan, automatic update, real time file protection, anti-phishing protection, USSD blocking, SMS/MMS scanner and filter, network protection, quarantine;
- Anti-theft – remote localization(GPS/network), remote wipe, remote lock, SMS or web interface for controlling anti-theft components, lock phone on SIM change, report thief's phone number, remote configuration, lock contacts and SMS/MMS, report thief's location on SIM changed, lock images and files;
- Authentication - lock screen with password protection, password policy (strength, length), maximum failure password attempts before wipe, inactivity timeout;
- Additional – data encryption, password protection for settings, no SIM activation, data network usage monitor, local wipe.

All these security features can be combined in a single application (antivirus and security application), which cares for smartphone's security.

On the other hand, based on the main channels for mobile banking (SMS, mobile websites and mobile applications) and the fact that the web browser can be a major source of malware applications, spoofing websites (based on phishing) and targeted Trojans, we turn our attention to another aspect of security enhancement – mobile web browsers.

The goal of our work is to determine which of the existing mobile antivirus and security applications and mobile web browsers would help the users to increase the mobile banking security level in Android OS.

The main tasks we set are:

- To be made a comparative analysis of some of the existing mobile antivirus and security applications for Android smartphones;
- To be checked how the absence of mobile web browsers security indicators for Android smartphones is related to certain security attacks;
- To be determined which combination mobile antivirus and security application – mobile web browser hides the least risk for mobile banking security in the case of Android OS.

Mobile antivirus and security applications

The existing mobile antivirus and security applications, which can be used to enhance mobile banking security of Android smartphones, include a wide range of security features. Using the information from AV-Comparatives research [AV-Comparatives Organization, 2014], we have identified which security features are available as functionality in 14 mobile antivirus and security applications. The security features are those mentioned earlier in the introduction. For each existing feature the antivirus and security application earns 1 point. The features are grouped in four main categories. The maximum points that can be earned in each category are as follows: anti-malware (8 points), anti-theft (9 points), authentication (4 points) and additional (5 points). The results are presented in Table 1.

Table 1 clearly shows the leading mobile antivirus and security applications according to the existing security features: Avast! Mobile Security 3.0.7650, Tencent Mobile Manager 4.8.2 and Kaspersky Mobile Security 11.4.4.232.

Surely the presence of certain security features in mobile antivirus and security applications is not a guarantee for their optimum performance and risk reduction. To be more precise about the effectiveness of these applications is necessary to be conducted number of tests checking actual security features’ operation. But this is not the subject of this report and could be only an outline for future work.

Table 1. Assessment on the presence of security features in mobile antivirus and security applications

Mobile Security Features / *Antivirus and Security Applications*	Anti-malware (max 8 p.)	Anti-theft (max 9 p.)	Authentication (max 4 p.)	Additional (max 5 p.)	Total
AhnLab V3 Mobile 2.1.2.13	2.5	5.5	1	1	14.5
Avast! Mobile Security 3.0.7650	7	9	3	4	23
Bitdefender Mobile Security Premium 2.19.344	4	6	3	2	15
ESET Mobile Security 3.0.937.0-15	5.5	4.5	3	2	15
F-Secure Mobile Security 9.2.15183	3.5	6	4	2	15.5
Ikarus mobile.security 1.7.20	6	4.5	4	1	15.5
Kaspersky Mobile Security 11.4.4.232	8	6	3	4	21
Lookout Premium 8.17-8a39d3f	4.5	3.5	3	3	14
Qihoo 360 Antivirus 1.0.0	4.5	6	2	2	14.5
Quick Heal Total Security 2.00.021	5.5	6	1	4	16.5
Sophos Security and Antivirus 3.0.1154(7)	7	5.5	1	2	15.5
Tencent Mobile Manager 4.8.2	7	8	3	4	22
Trend Micro Mobile Security 5.0	3.5	3	3	2	11.5
Webroot SecureAnywhere Mobile Complete 3.6.0.6610	5.5	4.5	3	0	13

Another important problem in security implementation and in antivirus and security application's usage is the user requirement to receive reliable security without hardening in any way the operation of the smartphone. This defines three more factors that must be considered when choosing a mobile antivirus and security application:

- License – free or commercial;
- Usability – easiness and intuitive usage, since users tend to avoid the installation of such applications due to their complexity;
- Battery usage – the effect of constantly working mobile antivirus and security application on the battery life.

Table 2 presents the final assessment of mobile antivirus and security applications, based on three factors – security features, usability and battery usage. The maximum points that can be earned for each factor are as follows: security features (6 points), usability (6 points) and battery usage (6 points). The factor "license" is for information purpose. It is not graded and does not affect the final assessment. The data in column "Security features" is extracted from Table 1 and the values are converted on six point basis. The data in columns "Usability" and "Battery usage" are derived respectively from report of AV-TEST research [AV-TEST Institute, 2014] and report of AV-Comparatives research [AV-Comparatives Organization, 2014].

The results in Table 2 show that the overall score of the assessed mobile antivirus and security applications is moving in close range, i.e. any of them can be chosen and the level of security will be also in close range. Among the commercial applications Kaspersky Mobile Security 11.4.4.232 is going to be a wise choice. And with regard to free license applications Tencent Mobile Manager 4.8.2 and Avast! Mobile Security 3.0.7650 are at the forefront.

Table 2. Final assessment of mobile antivirus and security applications

Factors Determining User's Choice and Security Applications / Mobile Antivirus	License	Security features (max 6 p.)	Usability (max 6 p.)	Battery usage (max 6 p.)	Total
AhnLab V3 Mobile 2.1.2.13	Free	3.3	6.0	6.0	15.3
Avast! Mobile Security 3.0.7650	Free/ Commercial	5.3	6.0	6.0	17.3
Bitdefender Mobile Security Premium 2.19.344	Commercial	3.5	6.0	6.0	15.5
ESET Mobile Security 3.0.937.0-15	Free/ Commercial	3.5	6.0	6.0	15.5
F-Secure Mobile Security 9.2.15183	Commercial	3.6	6.0	6.0	15.6
Ikarus mobile.security 1.7.20	Commercial	3.6	5.0	6.0	14.6
Kaspersky Mobile Security 11.4.4.232	Commercial	4.9	6.0	6.0	16.9
Lookout Premium 8.17-8a39d3f	Free/ Commercial	3.2	6.0	6.0	15.2
Qihoo 360 Antivirus 1.0.0	Free	3.3	6.0	6.0	15.3
Quick Heal Total Security 2.00.021	Commercial	3.8	5.5	6.0	15.3
Sophos Security and Antivirus 3.0.1154(7)	Free	3.6	6.0	6.0	15.6
Tencent Mobile Manager 4.8.2	Free	5.1	6.0	6.0	17.1
Trend Micro Mobile Security 5.0	Commercial	2.7	6.0	6.0	14.7
Webroot SecureAnywhere Mobile Complete 3.6.0.6610	Commercial	3.0	6.0	6.0	15

Mobile Web Browsers

Our main guideline for mobile web browser security will be directed to check if it is in compliance with The World Wide Web Consortium (W3C) user interface requirements and security indicators in particular [W3C, 2010]. If these requirements are not met and if some of the security indicators are missing, users can more easily be misled about the identity of the website or the security of the connection. This in turn is directly related to the mobile banking security, which may be compromised and exposed to attacks such as: phishing, malicious web sites, espionage, eavesdropping. The presence of the security indicators on a given web browser would not completely eliminate the risk, but users aware with the security indicators would make informed decisions about the websites that they visit.

User interface security indicators in web browsers are divided in two categories [Amrutkar, Traynor and Oorschot, 2011]:

- Primary user interface indicators – padlock icon, address bar, https URL prefix, favicon, site-identity button or URL coloring (signifying the presence of EV-SSL and SSL certificates);
- Secondary user interface indicators – security properties dialog, domain name, owner information, verifier information, information on why a certificate is trusted, validity period of manually accepted certificates (self-signed) and cipher details of an SSL connection.

Table 3 presents the results of a previously conducted research [Amrutkar, Traynor and Oorschot, 2012], which identifies how the absence of certain security indicators in particular mobile web browsers (for our study we have chosen only Android web browsers - Android 2.3.3, Chrome Beta 0.16.4130.199, Firefox Mobile 4 Beta 3, Opera Mini 6.0.24556, Opera Mobile 11.00) may lead to potential attacks such as phishing without SSL, phishing with SSL, phishing using a compromised CA (Certificate Authority) and industrial espionage/eavesdropping. Sign “+” implies that the corresponding attack is possible on the browser, while sign “-” implies that it is not possible.

Table 3. Potential attacks to mobile web browsers

Mobile Attacks Browsers	Phishing without SSL	Phishing with SSL	Phishing using a compromised CA	Industrial espionage/ Eavesdropping
Android 2.3.3	+	-	-	+
Chrome Beta 0.16.4130.199	-	-	-	+
Firefox Mobile 4 Beta 3	-	-	-	+
Opera Mini 6.0.24556	-	+	+	+
Opera Mobile 11.00	-	+	+	+

Phishing without SSL. If users accidentally enter a malicious website and have difficulties in viewing the entire website's URL due to the constrained screen size of the smartphone, they can be deceived by a domain name slightly different from the original one. Such website containing spoofed padlock icon and closely imitating legitimate website's content can easily provide an illusion of strong encryption. But if such website is rendered in a browser that offers identity information such as owner's name, it will be easier for the user to identify the phishing attack.

Phishing with SSL. If the attacker have decide not just to spoof padlock icon, but to buy a legitimate inexpensive SSL certificate for the website, the browser is going to display SSL security indicators such as https URL prefix and URL coloring site identity button in addition to the padlock icon providing illusion of security. In this case if the browser does not offer identity information the phishing attack will be successful.

Phishing using a compromised CA. When the attacker compromises CA, rogue certificates for legitimate website can be obtained. And if the CA is trusted by the browser, all certificates will be accepted and no warnings will be shown to the user. But if the browser offers user interface to enable certificate viewing, the user would not be so easily misled.

Industrial espionage/Eavesdropping. Such attacks can be achieved in browsers that do not show constantly https URL prefix, do not offer cipher details of an SSL connection, or are showing SSL indicators for websites with mixed content (for example the attacker can change website's code with a code injection).

Based on the data, presented in Table 3, we can make a conclusion that the users should be directed to the usage of either Chrome Beta 0.16.4130.199 or Firefox Mobile 4 Beta 3. Although they do not provide the maximum level of security, their usage is more appropriate than that of the other tested browsers.

Conclusion

The widespread usage of mobile phones and smartphones in particular leads to diversification of their functionalities. Mobile banking is a kind of additional feature, which provides many facilities to the users. One of the problems of its wider uptake is the question about security. In this study we attempt to facilitate users by representing them a comparison by certain criteria and offering them choices of mobile antivirus and security application and suitable mobile web browser. The focus was on Android OS, as it is first in the list of used mobile operating systems. According to the results, we can conclude that a very good option for consumers would be the usage of free Tencent Mobile Manager 4.8.2 or Avast! Mobile Security 3.0.7650 or commercial Kaspersky Mobile Security 11.4.4.232 in combination with mobile web browser Chrome Beta 0.16.4130.199 or Firefox Mobile 4 Beta 3. For sure it would not completely eliminate the risk to mobile banking security, but at least it would be a step in its enhancement and along with that would increase user's confidence in this type of service.

Bibliography

[Amrutkar, Traynor and Oorschot, 2011] C. Amrutkar, P. Traynor and P.C. Oorschot. An Empirical Evaluation of Security Indicators in Mobile Web Browsers, Georgia Institute of Technology, 2011

[Amrutkar, Traynor and Oorschot, 2012] C. Amrutkar, P. Traynor and P.C. Oorschot. Measuring SSL Indicators on Mobile Browsers: Extended Life, or End of the Road?, Information Security, 15th International Conference, ISC 2012, Passau, Germany, September 19-21, 2012. Proceedings, Springer Berlin Heidelberg, Berlin, ISBN: 978-3-642-33382-8

[AV-Comparatives Organization, 2014] AV-Comparatives Organization Web Site. Mobile Security Review, www.av-comparatives.org/wp-content/uploads/2014/09/avc_mob_201407_en.pdf, (Accessed 12 December 2014)

[AV-TEST Institute, 2012] AV-TEST Institute Web Site. Mobile Security Apps, www.av-test.org/fileadmin/pdf/publications/droidcon_2012_avtest_presentation_mobile_security_apps.pdf, (Accessed 12 December 2014)

[AV-TEST Institute, 2013] AV-TEST Institute Web Site. AV-TEST Examines 22 Antivirus Apps for Android Smartphones and Tablets, www.av-test.org/fileadmin/pdf/avtest_2013-01_android_testreport_english.pdf, (Accessed 12 December 2014)

[AV-TEST Institute, 2014] AV-TEST Institute Web Site. 32 Protection Apps for Android Put to the Test, www.av-test.org/en/news/news-single-view/32-protection-apps-for-android-put-to-the-test/?=, (Accessed 12 December 2014)

[Gartner Inc., 2014] Gartner Inc. Gartner Says Annual Smartphone Sales Surpassed Sales of Feature Phones for the First Time in 2013, www.gartner.com/newsroom/id/2665715, (Accessed 12 December 2014)

[McAfee Company, 2013] McAfee Company Web Site. Mobile Malware Growth Continuing in 2013, www.mcafee.com/us/security-awareness/articles/mobile-malware-growth-continuing-2013.aspx, (Accessed 12 December 2014)

[W3C, 2010] W3C Web site. Web Security Context: User Interface Guidelines, www.w3.org/TR/2010/WD-wsc-ui-20100309/, (Accessed 12 December 2014)

Authors' Information

Bonimir Penchev *– University of Economics – Varna, Assist. Prof., Varna, Bulgaria, Institute of Mathematics and Informatics - Bulgarian Academy of Sciences, PhD Student, Sofia, Bulgaria;*
e-mail: bonimir@gmail.com

Major Fields of Scientific Research: Information Systems Security, Mobile Banking

MULTIDIMENSIONAL ONTOLOGY OF ELECTRONIC DOCUMENT AS A BASE OF INFORMATION SYSTEM

Viacheslav Lanin

Abstract: *This work presents an approach based on unstructured information retrieving from electronic documents to design and organize information system functioning. Document processing is based on semantic indexing and additional meta-information including. Document model allows formalize intelligent document processing algorithms and integrate documents containing ontological resources. Semantic indexing is based on multidimensional ontology describing document semantics and structure. To process documents agent-based approach allowing to resolve task of including business logic into documents is supposed. A systems built on such principles will be more flexible, intelligent and adjustable to changing environment. The proposed approach lets to solve a wide range of tasks, assuming usage of electronic documents at all lifecycle steps, which in turn allows implement document-oriented paradigm of information system life cycle maintenance.*

Keywords: *ontology, electronic document, document model, information system, semantic indexing, intellectual agents.*

ACM Classification Keywords: *I. Computing Methodologies. I.2 ARTIFICIAL INTELLIGENCE: I.2.11 Distributed Artificial Intelligence – Intelligent agents; Multiagent systems. I.7 DOCUMENT AND TEXT PROCESSING: I.7.2 Document Preparation – Index generation.*

Introduction

There is a tendency in modern information systems to handle documents, i.e. unstructured data rather than structured ones. New system categories, e.g. social networks, enterprise information portals or wiki-resources, have become a pivotal part of modern information space. «Content», or «electronic document» as more general term, is a key component of such systems.

Nowadays it is supposed that an electronic document has metadata describing data structure and data semantics apart from the content itself. Due to this approach, electronic document processing can be organized in a brand new way, because fully-automated intelligent information analysis can be applied. One of the base components of the Semantic Web [Segaran, 2009] was developed using such approach, nonetheless this project in its current state is far from being fully implemented. However «Semantic Web» ideas can be realized within a framework of a single information system because of less scaling domains of interest. Now data needed to process documents are widely diluted, that means it is kept in documents itself as well as in databases of information systems created to handle that documents, and it appropriates only to a range of specific tasks being solved during the document life cycle. That is why there is a necessity to apply a uniform mechanism used to represent document information. One of the possible solutions is an ontological resource describing different aspects of an electronic document at different stages of its life cycle. This resource can be applied to resolve a variety of tasks connected with processing electronic documents by the means of information systems.

Document Model

To find a solution means to develop a model of electronic document, including metainformation, an ontological resource, which is a fundament for document content semantic indexing [Lukashevich, 2003], and document processing mechanism.

An electronic document is a set of structure elements further called fragments in the article. Most obvious examples of fragments are tables, headers, disposition form requisites etc. An electronic document can be represented as a set of four:

$$d = (S\ (F, R), C, o, M).$$

$S(F, R)$ – oriented hypergraph, which nodes (set F – a set of document fragments and edges set R – a set of relations between fragments), set C – document content, o – document ontology, M – mapping of set F into the o ontology concepts.

Hypergraph $S(F, R)$ assigns relations among documents fragments. Graph directivity is required to monitor such relations as *«is-a-part»* among fragments. Nodes are numbered, so it is clear how to determine a right order of fragments. Obviously, a graph edge including all nodes correspond to a whole document.

There can be two types of fragments: *simple fragments* are primary atomic elements like header or creation date and *compound fragments* consisting of simple ones. Formally, *fragment* is a set of two: *stat* – static fragment part including text, images, references, special symbols and some information required for fragment representation, *inf* – fragment part indicating its content place or a set of fragments.

Conventional document representation takes advantage of usual graphs, trees, for example, the XML tree-like description structure considerably facilitates document processing, but at the same time, it implies many notable restrictions. Hypergraph allows store arbitrary relations among fragments and sets of fragments.

Using notion described above, document template can be defined as $t = (S(F, R), C_0)$ where C_0 – primary content (standard template headers etc.).

To give the particular characteristics of problems solved here, let us make more specific o ontology definition:

$$o = (C, R, A),$$

where C – a set of ontology concepts, R – a set of relations among concepts, A – a set of axiomatic statements made upon this ontology.

Concepts can include both classes and its instances; axiomatic statements are used to determine rules and restrictions, which can not be expressed by means of relations.

To process documents it is necessary to implement an operation of an arbitrary part of a document detaching (range extracting operation), a set of graph nodes will be an input to this operation, and a subgraph induced by this set of nodes will be an output. Decipher operation is a mapping structure onto a fragment (graph node). Apart from document structure and content, visual and format document representation take an integral part, that is why, operation of a format document representation, i.e. a function determining correspondence among document fragments and a set of other fragments assigning representation rules is needed. The search operation is applied to different document elements: structure, content and representation, and search result will be document fragments relevant to search criteria.

Multidimensional Ontology of Electronic Documents

It is required to have consolidated knowledge about document structure and content (electronic document format and type, its structure) to resolve document processing tasks. All of these three aspects are presented in multidimensional ontology, but concepts from different aspects are interconnected. As a result, a single ontology of electronic documents is created. This resource has to maintain extension and enhancement configurable capabilities to solve newly appeared tasks at all life cycle stages of electronic documents.

Document Processing Logic

A problem of including data processing logic into electronic documents is being solved for many years (including commercial and research projects being implemented). There are commercial solutions such as office programming tools, macro programming languages; an example of context-based logic including are form filling automation tools (InfoPath, Google).

One of the main functions of all information systems is electronic document processing, however document is often considered just as «information containers» for users. Yet, if electronic documents are somehow «active», a range of conventions tasks for information systems will be solved more effectively. Document activation problem is supposed to be solved under the agent-based approach. Specific intelligent agents representing so-called «document interests» can be used to make documents more «dynamic». This approach allows not only make the intensity of tasks, being solved at all information system life cycle stages, lower, but also proposes new capabilities for developers and users.

The main idea of the approach supposes that each document has its intelligent agent containing all information about it and able to «Be of its interest» when solving diverse tasks. To implement intelligent document management it is necessary for an agent to have knowledge on document semantics and properly interpret document content. This in turn requires resolving semantic indexing task. Within the framework of the described approach basic domain knowledge is supposed to be stored in the ontology and it is accessible by the agents and can be interpreted by them.

Document life cycle maintenance

The proposed approach assumes to automate document life cycle in different information systems as well as to lower the labour intensity of domain analysis and domain changes during designing and maintaining information systems.

Based on semantic indexing and intelligent analysis of documents, a list of domain concepts, their attributes, restrictions, relations and operations can be derived. System analysts can take advantage of this information when developing a domain model:

- Information about changes in the content of documents in the process of information system exploitation can be applied while updating domain model,
- Information about changes in the document templates can be applied while updating document representation,
- Information about changes in the information system objects can be applied when generating and updating document content,
- Information about changes in business logic can be applied to change document content (for instance, manuals and instructions).

Solving tasks described above are based on the proposed approach of document activation by the means of intelligent processing tools.

Multi-agent semantic indexing algorithm

Simplifying the problem we assume that the first step of text analysis process was made (for instance using Yandex Mystem [Mystem, 2012]), i.e. a set of *morphological descriptors* for each word have been obtained. All other steps are performed by an agent-based semantic indexing. As it could be seen in fig. 1 syntax analysis is not used because it has high time complexity. Instead words order in a sentence is considered.

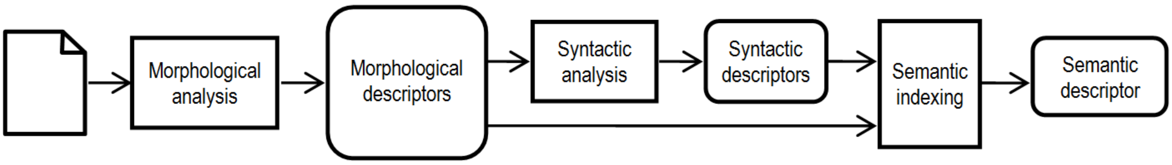


Figure 1. Steps of document analyses

The next step is a *semantic analysis*. The result of the semantic analysis is a semantic descriptor of plain text that binds the morphological descriptors to the elements of the domain ontology. Stop words are skipped.

The next step is a *structural analysis*. The structural analysis uses document’s structure, ontology that describes structure and semantic descriptors of plain text. At this step every concept of structural ontology tries to bind to the corresponding structural document element. The result of structural analysis is a semantic descriptor of whole text.

Descriptors (morphological, semantic) are a set of tags, which mark each word in the text.

Agent-based solution

Further, let us consider the process of building a semantic index based on multi-agent approach (see Fig. 2).

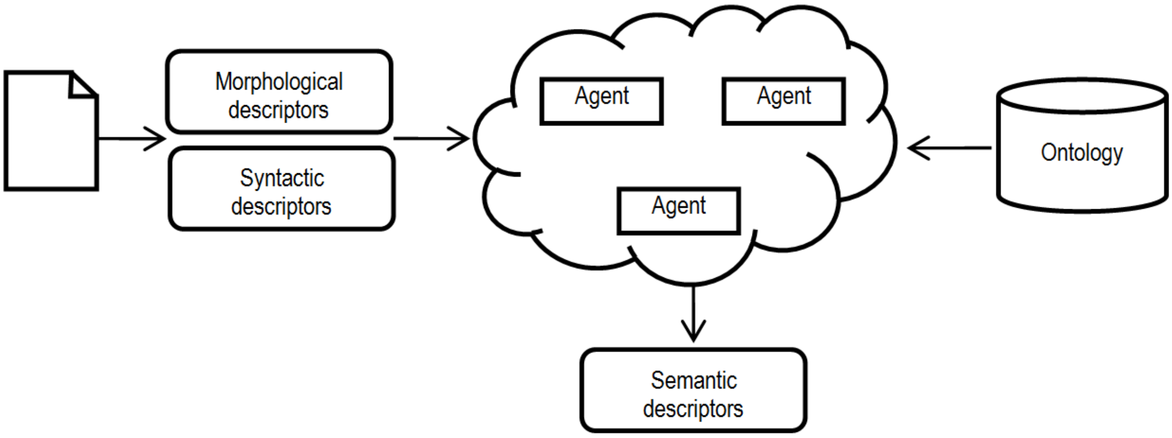


Figure 2. Architecture of agent platform

Agents have access to a domain ontology, structural ontology, morphological descriptors and electronic indexed documents. Indexing process is produced on the sentences in the text. Agents process sentences sequentially. The agents form a "team" to index the particular sentence. Thus, agents in the system are divided into teams after the start of the indexing.

Agent types

The following types of agents are identified in the system, according to the functional separation:

- Team Lead First Level Agent – TLFL agent,
- Team Lead Second Level Agent – TLSL agent,
- Word Indexer Agent – WI agent,

- Index Writer Agent – IW agent.

The task of WI agent is accessing the domain ontology and obtaining the set of possible semantic tags for the indexed word. An input word is passed to the WI agent for indexing with the parameters obtained at the stage of morphological analysis. The resulting set of possible semantic tags is passed to the TLSL agent.

TLSL agent binds to sentence morphological descriptors and distributes words to all available WI agents. TLSL agent finishes its work on the sentence when the consistent semantic descriptor is formed and written to the document. TLSL agent plans actions for the WI agents and participates in the auction for the resolution of contradictions. After building a consistent semantic descriptor TLSL agent transmits the generated semantic descriptor of the sentence to IW agent who writes semantic tags to the document.

TLFL agent binds to morphological descriptors of the document and distributes descriptors of the sentences to all available TLSL agents. TLFL agent monitors the TLSL agents work. If the work on the sentence is completed TLSL agent gives TLFL agent a new sentence. In addition, TLFL agent conducts an auction among TLSL agents to resolve ambiguity in the descriptors (see details in section «Agent negotiation»). Besides TLSL agents perform structural analysis. They distribute parts of structural ontology to TLSL agents.

Agent communication

Agents communicate through language FIPA ACL (Agent Communication Language developed by FIPA). Two types of actions are used. They inform (inform about anything) and perform (execution of an action).

Inform action type is implemented in the following cases:

- WI agent informs the TLSL agent about the completion of indexing word and give it the set of possible semantic tags; the content of the communication is as follows: (*id*, *tags*), where the *id* is the identifier word that come to be indexed, *tags* are returned set of possible semantic tags;
- TLSL agent informs the TLFL agent about a completion of an indexical sentence with a specific identifier; the content of this message contains an identifier of indexed sentence.

Perform action type is implemented in the following cases:

- TLFL agent gives to the TLSL agent a task to index a sentence with a specific descriptor; content will be like this: (*id*, *descriptor*), where the *id* is the identifier of the sentence, *descriptor* is descriptor of the sentence received as a result of syntactic and semantic analysis;

- TLSL agent gives a task to the WI agent to index a word with specific *id*; content will look like this: (*id*, *word*, *parameters*), where *id* is *ID* of the word, *word* is the word for indexing, *parameters* are parameters obtained at the stage of morphological and syntactic analysis;
- TLSL agent gives a task to the IW agent to write semantic tag of specific word; content is as follows: (*word*, *tag*), where *word* is an indexed word, *tag* is just a semantic tag of indexed word.

Planning

The planning is dynamic. TLSL agents themselves form a team of agents from the available WI agents. A count of needed WI agents depends on a sentence structure. If there is a lack of WI agents at the team formation time the TLSL agent may designate to perform indexing of few words at once to the same WI agent. TLFL agent monitors the performance of TLSL agents work and if they are released it assigns them new sentences for indexing. Completing of agents (WI and TLSL) work is monitored not only by sending their corresponding messages of inform type, but also by changing their states (agent states) in the meaning of "vacant."

Agent knowledge bases

WI agents and IW agents are primitive reflex agents working in the mode of stimulus-response. Their main function is a simple, no inference, execution of work. There are only procedural steps in the knowledge bases of these agents.

Knowledge bases of TLFL and TLSL agents represent productions with embedded procedural actions. In fact, the script actions are necessary for the distribution of work between agents. Accordingly TLSL agent knowledge base contains a script for word distribution among WI agents, and TLFL agent knowledge base includes a script for sentences distribution between agents TLSL.

Agent negotiation

TLFL agent conducts an auction among agents TLSL, each of which has a contextual memory (training component). Every TLSL agent using the contextual memory votes for a one option of sematic descriptor of the sentence. Option of semantic descriptor of the sentence with the highest number of votes will be considered as a true semantic descriptor of the sentence. The set of all consistent semantic descriptors of the sentences form the document semantic descriptor.

RELATED WORKS

Existing Document Ontology

Dublin core [Dublin core] is a set of metadata used to describe documents of various types (publications, audio records, video records). This set specification has status of official international standard (ISO: 15836 2003). The standard has two levels: Simple, comprising 15 elements and Qualified having three additional elements and element refinements (or qualifiers), which refine semantics of the elements. The main feature of Dublin Core is that every element is optional and might be repeated. Dublin Core is a powerful instrument used to describe resources of various types. The fact that it is widespread and flexible is its overwhelming advantage. However, it describes documents tags, i.e. information having indirect correlation with the document content. In this case it is impossible to describe other aspects of the electronic document.

Project ontologies «docOnto» [CNXML] developed by German research group KWARC (Knowledge Adaptation and Reasoning for Content) differ from other projects oriented on formal structure description development (CNXML document ontology) and document semantics (OMDoc document ontology). Members of this group also develop mechanisms of semantic document indexing and tools for document processing. CNXML document ontology (Connexions Markup Language) describes such terms as paragraph, section, reference etc. Ontology is formalized on UML. It gives detailed description of the document. Unfortunately, work in this direction is frozen, last changes date back 2007. One more direction in document ontologies creation is semantics description of documents for narrow subjects, where documents are well formalized, for example mathematical OMDoc documents. Mathematical Terms, theorems and several other terms are included in ontology.

Document ontology SHOE [SHOE] describes most types of documents. Academic papers are given particular emphasis. Dublin Core reference books and Document Classifier PubMed were the resource.

Document Ontology of Research Centre Linked Data DERI is developed by scholars of Irish Institute DERI (Digital Enterprise Research Institute) and is described in RDFS and OWL-DL. Terms referring project activity documentation are given in the ontology. Developers purposefully refused modelling structure and document content to accommodate flexibility and interoperability.

Muninn project document ontology became the result of processing archive documents of the First World War within the project Muninn WW1 [Muninn]. The Ontology describes bibliography, origins and storage description of the digital item. Most ontology classes are child classes of FOAF. That decision was compatibility possible, on the other hand, make adding additional features of document processing possible, i.e. features for representation document pages, copyright description, etc. One of the main ontology classes is Document, which is integrate class of FOAF Document and Creative Commons Works. Page class describes document pages, in its turn, Image class describes digital page image. Description of different document aspects, document structure in particular, is a significant benefit of this ontology. However, structure description is initially oriented on digital images of archive documents.

Each listed above document ontology has its advantages and disadvantages. We create our own ontology specialized on academic paper description.

System for creating text-based ontology

Nowadays there are some information systems that let you create text-based ontology models of documents or let you define correspondence of ontology models thereby transform one model into another one. We found two web-resources that let you create ontologies: OwlExporter and OntoGrid.

The core idea of OwlExporter is to take the annotations generated by an NLP pipeline and provide for a simple means of establishing a mapping between NLP (Natural Language Processing) and domain annotations on one hand and the concepts and relations of an existing NLP and domain-specific ontology on the other hand. Than the former can be automatically exported to the ontology in form of individuals and latter as data type or object properties [Witte] .

The resulting, populated ontology can be used then within any ontology-enabled tool for further querying, reasoning, visualization, or other processing.

OntoGrid is an instrumental system for automation of creating domain ontology using Grid-technologies and text analysis in natural language.

This system has bilingual linguistic processor for retrieving data from text in natural language. Worth derivational dictionary is used as a base for morphological analysis. It contains more than 3.2 million word forms. The index-linking process consists of 200 rules. “Key dictionary” is determined by words allocation analysis in text. The developers came up with new approach of revealing super phrase unities that consist of specific lexical units. The building of semantic net is carried out this way: the text is analyzed using text analysis system, semantic Q-nets are used as formal description of text meaning. The linguistic knowledge base of text analysis system is set of simple and complex word-groups of the domain. This base can be divided into simple-relation-realization base and critical-fragment-set, that let you determine which ontology elements are considered in this text. The next step is to create and develop the ontology in the context of GRID-net. A well-known OWL-standard is used to draw the ontology structure.

Conclusion

The proposed model allows giving a formal definition of intelligent processing of electronic documents, providing a wide range of opportunities to integrate documents with ontological resources. The agent-based approach enables to resolve a problem of adding business logic into documents. In contrast to other approaches, in this case no additional programming code or attributes should be added to documents. So, it is possible to abstract from technological processing particularities and format specifications. Such a system will be more flexible, intelligent and adjustable to changing environment. A proposed approach lets solve a wide range of tasks connected with usage of electronic documents at all lifecycle steps, that in turn will allow to implement document-oriented paradigm of information system life cycle maintenance.

Acknowledgement

The reported study was supported by RFBR, research project No. 14-07-31273.

Bibliography

[Segaran, 2009] Segaran T., Evans C., Taylor J. Programming the Semantic Web, O'Reilly Media, 2009.

[Lukashevich, 2003] Lukashevich N.V., Dobrov B.V. Bilingual information retrieval based on the automatic conceptual indexing // Computational linguistics and intelligent technologies. Proceedings of the International Conference “Dialogue-2003”. Protvino. June 11-16 2003y. / Ed. by I.M.Kobozevoy, N.I.Laufer, V.P.Selegeya - M.: Science, 2003. - pp.425-432.

[CNXML] CNXML/DocumentOntology http://mathweb.org/wiki/CNXML/DocumentOntology

[Dublin] Dublin Core Metadata Element Set, Version 1.1 http://dublincore.org/documents/dces/

[SHOE] Document Ontology (draft) http://www.cs.umd.edu/projects/plus/SHOE/ onts/docmnt1.0.html

[Muninn] Muninn Documents Ontology http://rdf.muninn-project.org/ontologies/documents.html

[Bessonov, 2012] Bessonov V., Lanin V.A, Sokolov G. A semantic indexing of electronic documents in open formats/INFORMATION THEORIES & APPLICATIONS, 2012, P. 139-148

[Mystem, 2012] Program for morphological analysis of text in Russian “Mystem”. [Electronic resource] [Mode of access:http://company.yandex.ru/technologies/mystem/] [Checked at: 24.06.12]

[Witte] Witte R., Khamis N., Rilling J., Flexible Ontology Population from Text: The OwlExporter Dept. of Comp. Science and Software Eng. Concordia University, Montreal, Canada. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2010/pdf/932_Paper.pdf

Authors' Information

Viacheslav Lanin *– National Research University Higher School of Economics, Department of Business Informatics; Russia, Perm, 614070, Studencheskaya st., 38; e-mail: lanin@perm.ru.*

Major Fields of Scientific Research: Intelligent agents, Ontologies, Document processing.

МОДЕЛЬ КОМБИНИРОВАННОЙ КАСКАДНОЙ РАДИАЛЬНО БАЗИСНОЙ НЕЙРОННОЙ СЕТИ И АЛГОРИТМ ЕЕ ОБУЧЕНИЯ

Елена Зайченко, Екатерина Малышевская

***Аннотация**: В статье предложена модель комбинированной каскадной радиально-базисной нейронной сети и представлен гибридный метод ее обучения, который заключается в том что процесс нахождения оптимальных значений параметров сети разбивается на 2 этапа, на первом из которых, оптимизируется нелинейные параметры радиально базисних функций, а на втором весы связей нейронной сети.*

***Ключевые слова**: нейронные сети, каскадные структуры нейронных сетей, радиально-базисные функции, методы оптимизации.*

***ACM Classification Keywords**: I.2.6 Connectionism and neural nets.*

Введение

Каскадные нейронные сети являются перспективным направленим в развитии нейронных сетей, потому разработка их мат ематических моделей и алгоритмов является актуальной задачей в сфере искусственного интеллекта. В статье предложен гибридный метод обучения каскадных нейронных сетей, который заключается в том что процесс нахождения оптимальных значений параметров каскадной радиально - базисной нейронной сети разбивается на 2 этапа, на первом из которых, оптимизируется нелинейные параметры радиально базисних функций, а на втором веса связей нейронной сети, благодаря чему снижается общая размерность задачи оптимизации и ускоряется работа алгоритма обучения.

Модель и алгоритм обучения комбинированной каскадной радиально базисной нейронной сети

Модель комбинированной каскадной радиально базисной нейронной сети (КРБНС) можно представить в таком виде:

1-й каскад.

$$\varphi_{1j}^{(1)}\left(x_j\right)=e^{\frac{-\left(x_j-c_{1j}\right)^2}{2r_1^2}}$$

$$\varphi_{kj}^{(1)}\left(x_j\right)=e^{\frac{-\left(x_j-c_{kj}\right)^2}{2r_k^2}}$$

Выходы РБФ нейрона первого каскада:

$$z_1^1=\prod_{j=1}^{n}\varphi_{1j}\left(x_j\right)=e^{-\sum_{j=1}^{n}\frac{\left(x_j-c_{1j}\right)^2}{2r_1^2}} \qquad (1)$$

$$z_2=\prod_{j=1}^{n}\varphi_{2j}\left(x_j\right)=e^{-\sum_{j=1}^{n}\frac{\left(x_j-c_{2j}\right)^2}{2r_2^2}}$$

$$z_k=\prod_{j=1}^{n}\varphi_{kj}\left(x_j\right)=e^{-\sum_{j=1}^{n}\frac{\left(x_j-c_{kj}\right)^2}{2r_k^2}}$$

Выходы нейронной сети первого каскада

$$y_1^{(1)}=\sum_{k=1}^{N}z_k^{(1)}w_{k1}^{(1)}=Z^TW_1$$

$$y_i^{(1)}=\sum_{k=1}^{N}z_k^{(1)}w_{ki}^{(1)}=Z^TW_i,\ i=\overline{1,N}$$

Вектор выходов $y^{(1)} = \left[y_i^{(1)} \right] = W^{(1)} Z$

$$W^{(1)} = \left\| w_{ik}^{(1)} \right\| \begin{matrix} k = \overline{1, N} \\ i = \overline{1, N} \end{matrix}$$

2-й каскад.

$$z_1^{(2)} = \exp\left(-\left\{ \sum_{j=1}^{n} \frac{\left(x_j - c_{1i}^{(2)} \right)^2}{2} + \sum_{i=1}^{N} \frac{\left(z_i^{(1)} - c_{i1}^{(1)} \right)^2}{2 r_{21}^{(2)}} \right\} \right)$$

$$z_k^{(2)} = \exp\left(-\left\{ \sum_{j=1}^{n} \frac{\left(x_j - c_{kj}^{(2)} \right)^2}{2 r_j^2} + \sum_{i=1}^{N} \frac{\left(z_i^{(1)} - c_{ik}^{(1)} \right)^2}{2 r_{2k}^2} \right\} \right)$$

$$y_k^{(2)} = y_k^{(1)} + \sum_{i=1}^{N} z_{ik}^{(2)} w_{ik}^{(2)}$$

***k*-й каскад.**

Непосредственные выходы *k*-го каскада:

$$\varphi_j^{(k)}\left(x_j \right) = e^{-\frac{\left(x_j - c_j^{(k)} \right)^2}{2 r_{jk}^2}}, \ j = \overline{1, n}$$

Входы с выходов предыдущих каскадов

$$\varphi^{(k)}\left(z_i^{(k)} \right) = \exp\left\{ -\frac{\left(z_i^{(k)} - c_i^{(k)} \right)^2}{2 r_k^2} \right\}, \ i = \overline{1, N}, \ k = \overline{1, K-1}$$

i-й выход *K*-го каскада (промежуточный)

$$z_i^{(K)} = \prod_{j=1}^{n} \varphi_j^{(k)}\left(x_j \right) \prod_{k=1}^{K-1} \varphi\left(z_k \right) = \exp\left\{ -\sum_{j=1}^{n} \frac{\left(x_j - c_j^{(k)} \right)}{2 r_j^2} - \sum_{k=1}^{K-1} \frac{\left(z_i^{(k)} - c_i^{(k)} \right)^2}{2 r_{ik}^2} \right\}$$

Общий выход после *K* каскадов

$$y_1^{(K)} = \sum_{k=1}^{K}\sum_{l=1}^{N} z_l^{(k)} w_{l1}^{(k)} = y_1^{(K-1)} + \sum_{l=1}^{N} z_l^{(K)} w_{l1}^{(K)}$$

$$y_i^{(K)} = \sum_{k=1}^{K}\sum_{l=1}^{N} z_l^{(k)} w_{li}^{(k)} = y_i^{(K-1)} + \sum_{l=1}^{N} z_l^{(K)} w_{li}^{(K)}$$, *i* – общий выход

Или в матричной записи

$$Y_1^{(k)} = W_1^{(k)} Z^{(k)},\ Y_i^{(k)} = W_i^{(k)} Z^{(k)},\ i = \overline{1,N}$$

Модель описания работы каскадной РБФН сети.

1. Модель 1-го каскада.

Используются РБ-функции для всех входов (входы $j = \overline{1,n}$; выходы $i = \overline{1,N}$):

$\varphi_{1j}^{(1)}(x_j) = e^{-\frac{(x_j - c_{1j})^2}{2r_1^2}}$, $i = \overline{1,N}$ где N = 6 – число выходов сети.

Первый z:

$$\varphi_{kj}^{(1)}(x_j) = e^{-\frac{(x_j - c_{kj})^2}{2r_k^2}} \qquad (2)$$

Первый выход РБФ нейрона 1-го каскада:

$$z_1^{(1)} = \prod_{j=1}^{n} \varphi_{1j}^{(1)}(x_j) = \exp\left\{\sum_{j=1}^{n} -\frac{(x_j - c_{1j})^2}{2r_1^2}\right\},$$

i-й выход

$$z_k^{(1)} = \prod_{j=1}^{n} \varphi_{kj}^{(1)}(x_j) = \exp\left\{\sum_{j=1}^{n} -\frac{(x_j - c_{kj})^2}{2r_k^2}\right\}$$

Выходы сети:

$$y_1^{(1)} = \sum_{k=1}^{N} z_k^{(1)} w_{k1}^{(1)}\ ,\ y_i^{(1)} = \sum_{k=1}^{N} z_k^{(1)} w_{ki}^{(1)}\ ,\ y_N^{(1)} = \sum_{k=1}^{N} z_k^{(1)} w_{kN}^{(1)}$$

$$y^{(1)} = \left[y_i^{(1)} \right] = W^{(1)} Z$$

$$W^{(1)} = \left\| w_{ik} \right\| \begin{matrix} k = \overline{1, N} \\ i = \overline{1, N} \end{matrix}$$

2. Модель 2-го каскада.

Непосредственные входы:

$$\varphi_{1j}^{(2)}\left(x_j\right) = e^{-\frac{\left(x_j - c_{1j}\right)^2}{2 r_{1j}^2}}\ ,\ j = \overline{1, n}\ .$$

$$\varphi_{ij}^{(2)}\left(x_j\right) = e^{-\frac{\left(x_j - c_{ij}^{(2)}\right)^2}{2 r_{ij}^2}}\ ,\ j = \overline{1, n}\ ,\ i = \overline{1, N}\ .$$

Входы с выходов 1-го каскада:

$$\varphi\left(z_i^{(1)}\right) = e^{-\frac{\left(z_i^{(1)} - c_{n+i}^{(1)}\right)^2}{2 r_{n+i}^2}}$$

$$z_1^{(2)} = \prod_{j=1}^{n} \varphi_{1j}^{(2)}\left(x_j\right) \varphi\left(z_1^{(1)}\right) = \exp\left\{ - \sum_{j=1}^{n} \frac{\left(x_j - c_{1j}^{(2)}\right)^2}{2 r_{1j}^2} - \frac{\left(z_1^{(1)} - c_{1n+1}^{(1)}\right)^2}{2 r_{1n+1}^2} \right\}$$

$$z_i^{(2)} = \prod_{j=1}^{n} \varphi_{ij}^{(2)}\left(x_j\right) \varphi\left(z_i^{(1)}\right) = \exp\left\{ - \sum_{j=1}^{n} \frac{\left(x_j - c_{ij}^{(2)}\right)^2}{2 r_{ij}^2} - \frac{\left(z_i^{(1)} - c_{n+i}^{(1)}\right)^2}{2 r_{1n+i}^2} \right\}$$

Выходы 2-го каскада:

$$y_1^{(2)} = \sum_{i=1}^{N} z_i^{(1)} w_{1i}^{(1)} + \sum_{i=1}^{N} z_i^{(2)} w_{1i}^{(2)}\ ;\ y_k^{(2)} = \sum_{i=1}^{N} z_i^{(1)} w_{ki}^{(1)} + \sum_{i=1}^{N} z_i^{(2)} w_{ki}^{(2)}$$

Алгоритм обучения КРБНС.

Общий критерий обучения выглядит так:

Найти такие матрицы весов $\mathrm{W_k} = \left\| \mathrm{W_{il}^{(k)}} \right\|$, $\mathrm{i} = \overline{1,\mathrm{N}}$, $1 = \overline{1,\mathrm{N}}$, $\mathrm{k} = \overline{1,\mathrm{K}}$; а также параметры РБ-функций $\left\{\mathrm{c_{ij}^{(k)}}\right\}$ и $\left\{\mathrm{r_{ij}^{(k)}}\right\}$ для которых $E = \sum_{t=1}^{T} \sum_{i=1}^{N} \left(y_{it}^{\mathrm{target}} - \hat{y}_{it}^{(k)}(W,C,R)\right)^2 \to \min\limits_{W,C,R}$, где $\mathrm{y_{it}^{target}}$ – требуемый желаемый выход *i*-го класса, $\mathrm{\hat{y}_{it}}(W,C,R)$ – выход модели *i*-го класса для образца t, T – общий объем обучающей выборки.

Описание алгоритма.

Общий алгоритм состоит из последовательности итераций. Обучение проходит последовательно, начиная с первого каскада [Fahlman & Lebiere, 1989].

Алгоритм обучения каскада 1.

Используем гибридный алгоритм состоящий из последовательности двух этапов.

- 1 этап. Обучение весов связей $W_{il}^{(1)}$;
- 2 этап. Настройка параметров РБ-функций $\left\{r_{ij}, c_{ij}^{(k)}\right\}$.

Этап 1.

1. Первоначально выберем начальные значения параметров РБФ [Kovacevic & Loncaric, 1997] $c_{ij}^{(1)}(0)$ и $r_{ij}^{2}(0)$ их можно инициализировать случайно или рассчитать так $c_{ij}(0) = \frac{1}{n_i}\sum_{t\in V_i} x_{jt}$, где *t* – номер образца; *V_i* – подмножество образцов класса *i* ($\mathrm{i} = \overline{1,\mathrm{N}}$); n_i – число образцов класса V_i в обучающей выборке, x_{jt} – значение входа *j* образца *t*:

$$\mathrm{r^2}_{\mathrm{ij}}(0) = \frac{1}{n_i - 1}\sum_{t\in V_i}\left(x_{jt} - c_{ij}(0)\right)^2 .$$

2. Далее определяем $\left\{\varphi_{ij}^{(1)}\left(x_{jt}\right)\right\}, t \in T$ и находим $z_{it}^{(1)}$. Обозначим для удобства $\left[z_{it}^{(1)}\right]_{t\in V_i} = z_i^{(1)}$.

$y_{1t}^{(1)} = \sum_{i=1}^{N} z_{it}^{(1)} w_{i1}^{(t)}$, $y_{Nt}^{(1)} = \sum_{i=1}^{N} z_{it}^{(1)} w_{iN}^{(t)}$ – выходы 1-го каскада.

Поскольку классификация происходит по критерию max выхода, то решающее правило классификации $X_t = \left[x_{jt}^{'}\right]$, $\mathrm{j} = \overline{1,\mathrm{N}}$ относится к классу *V_i*, если $y_{it}^{(1)} = \sum_{l=1}^{N} z_{lt}^{(1)} w_{li}^{(1)} = \max_k y_{kt}^{(1)} = \max_k \sum_{l=1}^{N} z_{lt}^{(1)} w_{lk}^{(1)}$.

Обозначим желаемое значение *i*-го выхода (класса) $y_{\mathrm{target}\,i}^{\mathrm{t}} = y_{\max}$ образцов из класса V_i, а для всех образцов из других классов $\overline{V_i}$ через $y_{\mathrm{target}\,i}^{\mathrm{t}} = y_{\min}$. Запишем соответствующие неравенства для *i*–го выхода первого каскада

$$y_{it} = z^{(1)} w^{(1)} = \sum_{l=1}^{N} z_{lt}^{(1)} w_{li}^{(1)} \text{ , } \ge y_{\mathrm{target}}^{t} = y_{\max}, \quad t \in V_i \tag{3}$$

$$\le y_{\mathrm{target}}^{t} = y_{\min}, \quad t \notin V_i \tag{4}$$

Тогда можно записать следующую задачу ЛП

$$\min \sum_{t\in V_i} \varepsilon_t \tag{5}$$

при таких условиях:

$$\sum_{l=1}^{N} z_{lt}^{(1)} w_{li}^{(1)} - \varepsilon_t = y_{\max}, t \in V_i \text{, } i = \overline{1,N} \tag{6}$$

$$\sum_{l=1}^{N} z_{lt}^{(1)} w_{li}^{(1)} + \varepsilon_t = y_{\min}, t \notin V_i \text{, } i = \overline{1,N} \tag{7}$$

$$\varepsilon_t \ge 0$$

Решаем ЗЛП (5) – (7) симплекс методом и если она разрешима, то конец обучения 1 каскада, иначе если она оказалась неразрешимой по соответствующему признаку симплекс-метода, то переходим к следующей вспомогательной задаче:

$$\min \sum_{t=1}^{T} \varepsilon_t^2 \tag{8}$$

при условиях (6), (7). Эта задача квадратичного программирования, которая решается стандартным методом [Зайченко, 2004].

Повторяем ее решение для всех начальных значений весов $\left[W_{il}^{(1)}\right]_{l=\overline{1,N}}^{i=\overline{1,N}}$.

Конец этапа 1.

Переходим ко второму этапу, на котором оптимизируем значения параметров РБ-функций $c_{ij}^{(1)}$ и $r_{ij}^{(1)}$.

Этап 2

Оптимизируемый критерий имеет следующий вид:

$$E = \frac{1}{T}\sum_{i=1}^{N}\sum_{t\in V_i}\left(y_{it}^{\text{target}} - \hat{y}_{it}(c,r)\right)^2 \to \min \tag{9}$$

где $y_{it}^{\text{target}} = \begin{cases} y_{\max}, & \text{if } t \in V_i \\ y_{\min}, & \text{if } t \notin V_i \end{cases}$ - заданное значение для образца *t i*-го выхода, V_i— множество образцов i-го класса обучающей выборки, $\hat{y}_{it}$ – *i*-й выход для t-го образца КРБН-сети.

Для оптимизации параметров используется градиентный метод или метод обучения Відроу-Хоффа Найдем значение $\frac{\partial E}{\partial c_{ij}^{(1)}}$ для градиента:

$$\frac{\partial E}{\partial c_{ij}^{(1)}} = -\frac{1}{T}\sum_{t=1}^{T}\left(y_{it}^{\text{target}} - \hat{y}_{it}(c,r)\right)\frac{\partial \hat{y}_{it}}{\partial c_{ij}^{(1)}} = -\frac{1}{T}\sum_{t=1}^{T}\left(y_{it}^{\text{target}} - \hat{y}_{it}(c,r)\right)\frac{\partial \hat{y}_{it}}{\partial c_{ij}^{(1)}} \tag{10}$$

С учетом того, что:

$$\hat{y}_{it} = \sum_{l=1}^{N} z_{lt}^{(1)} w_{li}^{(1)} = \sum_{l=1}^{N} \prod_{j=1}^{n} \varphi\left(x_{lj}^{t}\right) w_{li}^{(1)} = \sum_{l=1}^{N} \exp\left(-\sum_{j=1}^{n} \frac{\left(x_{jt} - c_{lj}^{(1)}\right)^2}{2r_j^2} \right) w_{li}^{(1)}, \ j = \overline{1,n}, \ l = \overline{1,N} \quad (11)$$

$$\frac{\partial E}{\partial c_{lj}^{(1)}} = w_{ji} \exp\left(-\sum_{j=1}^{n} \frac{\left(x_{jt} - c_{lj}^{(1)}\right)^2}{2r_j^2} \right) \frac{\left(x_{jt} - c_{lj}^{(1)}\right)}{2r_j^2} \quad (12)$$

$$\frac{\partial E}{\partial r_j^{(1)}} = w_{ji} \exp\left(-\sum_{j=1}^{n} \frac{\left(x_{jt} - c_{lj}^{(1)}\right)^2}{2r_j^2} \right) \frac{\left(x_{jt} - c_{lj}^{(1)}\right)^2}{2r_j^3} \quad (13)$$

Далее реализуем градиентный алгоритм спуска и находим рекуррентно:

$$c_{lj}^{(1)}(m+1) = c_{lj}^{(1)}(m) - \gamma_m \frac{\partial E(m)}{\partial c_{lj}^{(1)}}, \ m = 1, 2, \ldots \quad (14)$$

$$r_{lj}^{(1)}(m+1) = r_{lj}^{(1)}(m) - \gamma_m' \frac{\partial E(m)}{\partial r_{lj}}, \ m = 1, 2, \ldots \quad (15)$$

Условия сходимости

а. $\gamma_m \to 0; \ m \to \infty$

б. $\sum_{m=0}^{\infty} \gamma_m = \infty$

в. $\sum_{m=0}^{\infty} \gamma_m^2 < \infty$

Итерации градиентного спуска повторяем до тех пор, пока значения $c_{lj}^{(1)}$ и $r_{lj}^{(1)}$ не будут стабилизированы.

В результате находим $c_{lj\,\mathrm{new}}^{(1)} = c_{lj}^{(1)}(1)$ и $r_{lj\,\mathrm{new}}^{(1)} = r_{lj}^{(1)}(1)$.

Далее переходим снова к этапу 2 и решаем (7) при условиях (5) и (6) с новыми значениями параметров РБ-функции $c_{lj\,\mathrm{new}}^{(1)}$ и $r_{lj\,\mathrm{new}}^{(1)}$.

Последовательность этапов 1 и 2 повторяем до тех пор, пока значения параметров $c_{lj}^{(1)}$ и $r_{lj}^{(1)}$ и веса $\left[w_{li}^{(1)}\right]_{i=\overline{1,N}}^{l=\overline{1,N}}$ не стабилизируются. На этом обучение параметров первого каскада заканчивается.

Поскольку число итераций градиентного метода зависит нелинейно от размерности (числа варьируемых параметров), то разбиение предлагаемого алгоритма оптимизации на 2 отдельных этапа существенно сокращает размерность задачи. Благодаря этому, общее число итераций предл. алгоритма значительно меньше, чем у классического градиентного метода, а его скорость сходимости выше.

Алгоритм обучения каскада 2

Далее переходим к оптимизации параметров обучения второго каскада (*k*=2) зафиксировав параметры РБФ и выходы каскада 1 ($y_1^{(1)},\dots,y_N^{(1)}$).

Для него используется тот же гибридный алгоритм что и ранее (для каскада 1).

При этом на этапе 1 для *i*-го выхода каскада 2 имеем:

$$y_{it}^{(2)} = \sum_{l=1}^{N} z_{lt}^{(2)} w_{li}^{(2)} + \sum_{l=1}^{N} z_{lt}^{(1)} w_{li}^{(1)} = \sum_{l=1}^{N} z_{lt}^{(2)} w_{li}^{(2)} + y_{it}^{(1)} \qquad (16)$$

Решаем задачу квадратичного программирования для i-го выхода:

$$\min \sum_{t \in V_i} \varepsilon_t^2 \qquad (17)$$

при условиях

$$\sum_{l=1}^{N} z_{lt}^{(2)} w_{li}^{(2)} + y_{it}^{(1)} - \varepsilon_t = y_{\max}, \quad t \in V_i \qquad (18)$$

$$\sum_{l=1}^{N} z_{lt}^{(2)} w_{li}^{(2)} + y_{it}^{(1)} + \varepsilon_t = y_{\min}, \quad t \notin V_i \qquad (19)$$

В результате находим начальные веса $\left[w_{li}^{(2)}\right]_{i=\overline{1,N}}^{l=\overline{1,N}}$, затем переходим ко второму этапу.

На втором этапе градиентным методом или методом спряженного градиента вычисляем значения параметров РБ-функций второго каскада по формулам, аналогичным (14), (15).

Повторив многократно 2 этапа, определяем установившиеся параметры РБ-функций и веса $\left[w_{li}^{(2)}\right]$ второго каскада.

Обозначим через E2 значение общего критерия после 2-го каскада.

Проверка условия останова.

Если а) $\left|E_2 - E_1\right| < \varepsilon$ или б) $E_2 > E_1$, то stop; синтез структуры заканчивается.

В противном случае, переходим к синтезу 3-го каскада.

Последовательность итераций останавливается на каскаде *k*, когда $E_{k+1} > E_k$ или $\left|E_{k+1} - E_k\right| < \varepsilon$, где ε- заданная точность.

Результаты экспериментов

Проводилось 2 эксперимента классификации типа и процентного содержания эпителия на шейке матки. Всего было 100 наблюдений. Каждое наблюдение это известный результат биопсии. Другими словами входы сети это часть изображения, где была взята биопсия и известны ее результаты, а выходы сети это тип эпителия, то есть результат биопсии. Выборка делилась на обучающую (80) и тренировочную (20) подвыборки. Для анализа результатов использовался cross validation.

В первом эксперименте мы классифицировали эпителий на 6 типов. Использовалось 4 вида нейронных сетей. Сравнивались результаты классификации типов эпителия известных нейронных сетей с новым алгоритмом комбинированной каскадной радиально базисной нейронной сети.

Во втором эксперименте мы провеяли область на наличие опасного типа эпителия. Так как опасными считаются определенные три типа ткани - мы определяли их суммарный процент на исследуемой области органа. Для того чтоб этого сделать мы фактически провели первый эксперимент с пост процессингом: мы использовали наблюдения с полными результатами биопсии (включая процентное содержание каждого типа эпителия) и проверили результаты для суммарного процента трех опасных видов эпителия.

Результаты приведены в следующей Таблице 1. Из таблицы видно, что комбинированный алгоритм каскадной нейронной сети дает лучший результат.

Таблица 1. Результаты классификации типов тканей (СКО)

	Каскадная нейронная сеть	Нейронная сеть Back propagation	Радиально базисная нейронная сеть	Каскадная радиально базисная нейронная сеть
6 типов тканей	0.0479	0.0584	0.0610	0.0361
CIN1+CIN2+CIN3 (до обучения)	0.0832	0.1089	0.0569	0.0498
CIN1+CIN2+CIN3 (после обучения)	0.0479	0.0584	0.0610	0.0361

Заключение

Разработан новый метод синтеза нейронных сетей на основе которого получена новая структура комбинированной каскадной радиально-базисной нейронной сети для обработки оптических изображений.

Разработан метод обучения каскадной радиально-базисной нейронной сети, который заключается в том, что процесс нахождения оптимальных значений параметров КРБНМ разбивается на 2 этапа, на первом из которых, оптимизируется нелинейные параметры РБ функций, а на втором весы связей НМ, благодаря чему снижается общая размерность задачи оптимизации и ускоряется работа алгоритма обучения.

Из проведенных экспериментальных исследований видно, что комбинированная нейронная сеть дает лучший результат классификации.

Библиография

[Fahlman & Lebiere, 1989] Scott E. Fahlman, Christian Lebiere, “The Cascade-Correlation Learning Architecture”, NIPS 1989, pp. 524-532

[Kovacevic & Loncaric, 1997] Domagoj Kovacevic, Sven Loncaric, “Radial Basis Function-based Image Segmentation using a Receptive Field”, Computer-Based Medical Systems, Proceedings, Tenth IEEE Symposium, 1997, pp. 126-130

[Зайченко, 2004] Зайченко Ю.П., “Основы проэктирования интеллектуальных систем”, Учебное пособие – К.: Издательский дом “Слово”, 2004.

Сведения об авторах

Елена Юрьевна Зайченко *– НТУУ „КПИ”, д.т.н., профессор, Киев-03056, Украина; e-mail: syncmaster@bigmir.net*

Основные области научных исследований: методы оптимизации, нейронные сети, компьютерные сети

Екатерина Николаевна Малышевская *- НТУУ „КПИ”, аспирант, Киев-03056, Украина; e-mail: kate.inv@gmail.com*

Основные области научных исследований: нейронные сети, обработка изображений.

The Model for the Combined Cascade Radial Basis Neural Network and Its Learning Algorithm

Olena Zaychenko, Kateryna Malyshevska

Abstract*: In this article, the combined cascade radial basis network is proposed and a hybrid learning method is presented, in which the optimal values for network’s parameters are calculated in two stages. In the first stage, the nonlinear parameters of radial basis functions are optimized and, in the second stage, the connection weights are optimized.*

Keywords*: radial basis function network, neural network, cascade neural network, learning algorithm, neural network model.*

A SURVEY OF MATHEMATICAL AND INFORMATIONAL FOUNDATIONS OF THE BIGARM ACCESS METHOD

Krassimira Ivanova

Abstract: *The BigArM is an access method for storing and accessing Big Data. It is under development. In this survey we present its mathematical and informational foundations as well as its requirements to realization characteristics. Firstly, we outline the needed basic mathematical concepts, the Names Sets, and hierarchies of named sets aimed to create a specialized model for organization of information bases called "Multi-Domain Information Model" (MDIM). The "Information Spaces" defined in the model are kind of strong hierarchies of enumerations (named sets). Further we remember the main features of hashing and types of hash tables as well as the idea of "Dynamic perfect hashing" and "Trie", especially – the "Burst trie". Hash tables and tries give very good starting point. The main problem is that they are designed as structures in the main memory which has limited size, especially in small desktop and laptop computers. To solve this problem, dynamic perfect hashing and burst tries will be realized as external memory structures in BigArM.*

Keywords*: BigArM, Big Data, Cloud computing.*

ACM Keywords: *D.4.3 File Systems Management, Access methods.*

Introduction

In this survey we will present the mathematical and informational foundations of a new generation of the access methods based on numbered information spaces [Markov, 1984; Markov, 2004]. Firstly we will outline the needed basic mathematical concepts, the Names Sets, and hierarchies of named sets aimed to create a specialized mathematical model for organization of information bases called "Multi-Domain Information Model" (MDIM). The "Information Spaces" defined in the model are kind of strong hierarchies of enumerations (named sets). Further we will remember the main features of hashing and types of hash tables as well as the idea of "Dynamic perfect hashing" and "Trie", especially – the "Burst trie". Hash tables and tries give very good starting point. The main problem is that they are designed as structures in the main memory which has limited size, especially in small desktop and laptop computers. To solve this problem, in BigArM dynamic perfect hashing and burst tries will be realized as external memory structures.

Basic mathematical concepts

Let remember the some basic mathematical concepts needed for this research [Bourbaki, 1960; Burgin, 2010].

$\varnothing$ is the *empty set*.

If X is a set, then $r \in X$ means that r belongs to X or r is a member of X. If X and Y are sets, then $Y \subseteq X$ means that Y is a subset of X, i.e., Y is a set such that all elements of Y belong to X.

The *union* $Y \cup X$ of two sets Y and X is the set that consists of all elements from Y and from X.

The *intersection* $Y \cap X$ of two sets Y and X is the set that consists of all elements that belong both to Y and to X.

The *union* $\bigcup_{i\in I} X_i$ of sets X_i is the set that consists of all elements from all sets X_i, $i\in I$.

The *intersection* $\bigcap_{i\in I} X_i$ of sets X_i is the set that consists of all elements that belong to each set X_i, $i\in I$.

The *difference* $Y\backslash X$ of two sets Y and X is the set that consists of all elements that belong to Y but does not belong to X.

If X is a set, then 2^X is the *power set* of X, which consists of all subsets of X. The power set of X is also denoted by $\boldsymbol{P}(X)$.

If X and Y are sets, then $X \times Y = \{(x, y); x \in X, y \in Y\}$ is the direct or Cartesian product of X and Y, in other words, $X \times Y$ is the set of all pairs (x, y), in which x belongs to X and y belongs to Y.

Elements of the set X^n have the form $(x_1, x_2, \ldots, x_n)$ with all $x_i \in X$ and are called *n*-tuples, or simply, tuples.

A fundamental structure of mathematics is *function*. However, functions are special kinds of binary relations between two sets.

A *binary relation* T between sets X and Y is a subset of the direct product $X \times Y$. The set X is called the *domain* of T ($X = \mathrm{Dom}(T)$) and Y is called the *codomain* of T ($Y = \mathrm{CD}(T)$). The *range* of the relation T is $\mathrm{Rg}(T) = \{y; \exists\, x \in X\, ((x, y) \in T)\}$.

The *domain of definition* of the relation T is $\mathrm{DDom}(T) = \{x; \exists\, y \in Y\, ((x, y) \in T)\}$. If $(x, y) \in T$, then one says that the elements x and y are in relation T, and one also writes $T(x, y)$.

Binary relations are also called multivalued functions (mappings or maps).

Y^X is the set of all mappings from X into Y.

$$X^n = \underbrace{X \times X \times \ldots X \times X}_{n}.$$

A *function* (also called a *mapping* or *map* or *total function* or *total mapping*) *f* from *X* to *Y* is a binary relation between sets *X* and *Y* in which:

- There are no elements from X which are corresponded to more than one element from Y;
- To any element from X, some element from Y is corresponded.

Often total functions are also called everywhere defined functions. Traditionally, the element *f*(*a*) is called the image of the element *a* and denotes the value of *f* on the element *a* from *X*. At the same time, the function *f* is also denoted by *f*: *X* → Y or by *f*(*x*). In the latter formula, *x* is a variable and not a concrete element from *X*.

A *partial function* (or *partial mapping*) *f* from *X* to *Y* is a binary relation between sets *X* and *Y* in which there are no elements from *X* which are corresponded to more than one element from *Y*.

Thus, any function is also a partial function. Sometimes, when the domain of a partial function is not specified, we call it simply a function because any partial function is a total function on its domain.

A *multivalued function* (or *mapping*) *f* from *X* to *Y* is any binary relation between sets *X* and *Y*.

$f(x) \equiv a$ means that the function *f*(*x*) is equal to *a* at all points where *f*(*x*) is defined.

Two important concepts of mathematics are the domain and range of a function. However, there is some ambiguity for the first of them. Namely, there are two distinct meanings in current mathematical usage for this concept. In the majority of mathematical areas, including the calculus and analysis, the term "domain of *f*" is used for the set of all values *x* such that *f*(*x*) is defined. However, some mathematicians (in particular, category theorists), consider the domain of a function *f*: *X*→*Y* to be *X*, irrespective of whether *f*(*x*) is defined for all *x* in *X*. To eliminate this ambiguity, we suggest the following terminology consistent with the current practice in mathematics.

If *f* is a function from *X* into *Y*, then the set *X* is called the *domain* of *f* (it is denoted by Dom*f*) and *Y* is called the *codomain* of *T* (it is denoted by Codom*f*). The *range* Rg*f* of the function *f* is the set of all elements from *Y* assigned by *f* to, at least, one element from *X*, or formally, $\mathrm{Rg}f = \{y;\ \exists\, x \in X\ (f(x) = y)\}$. The *domain of definition* DDom*f* of the function *f* is the set of all elements from *X* that related by *f* to, at least, one element from *Y* is or formally, $\mathrm{DDom}f = \{x;\ \exists\, y \in Y\ (f(x) = y)\}$. Thus, for a partial function *f*(*x*), its domain of definition DDom*f* is the set of all elements for which *f*(*x*) is defined.

Taking two mappings (functions) f: $X \to Y$ and g: $Y \to Z$, it is possible to build a new mapping (function) gf: $X \to Z$ that is called *composition* or *superposition* of mappings (functions) f and g and defined by the rule $gf(x) = g(f(x))$ for all x from X.

An n-ary relation R in a set X is a subset of the n^{th} power of X, i.e., $R \subseteq X^n$. If $(a_1, a_2, \ldots, a_n) \in R$, then one says that the elements $a_1, a_2, \ldots, a_n$ from X are in relation R.

Named sets

The concept “Named set” was defined by Mark Burgin. Here we will follow [Burgin, 2011].

Named set **X** is a triple **X** = (X, μ, I) where:

- X is the *support* of **X** and is denoted by S(**X**);
- I is the *component of names* (also called *set of names* or *reflector*) of **X** and is denoted by N(**X**);
- μ: $X \to I$ is the *naming map* or *naming correspondence* (also called *reflection*) of the named set **X** and is denoted by n(**X**).

The most popular type of named sets is a named set **X** = (X, μ, I) in which X and I are sets and μ consists of connections between their elements. When these connections are set theoretical, i.e., each connection is represented by a pair (x, a) where x is an element from X and a is its name from I, we have a *set theoretical named set*, which is binary relation.

A name $a \in I$ is called *empty* if $\mu^{-1}(a) = \varnothing$.

A named set **X** is called:

- *Normalized* if in **X** there are no empty names;
- *Conormalized* if in **X** there no elements without names;

Named sets as special cases include:

- Usual sets;
- Fuzzy sets;
- Multisets;
- Enumerations;
- Sequences (countable as well as uncountable)

etc.

A lot of examples of named sets we may find in linguistics studying semantical aspects that are connected with applying different elements of language (words, phrases, texts) to their meaning [Burgin & Gladun, 1989; Burgin, 2011].

A named set **Y** = (Y, η, J) is called *named subset* of named set **X** if $Y \subseteq X$, $J \subseteq I$, and $\eta = \mu |_{(Y,J)}$ ($\eta \subseteq \mu \cap (Y \times J)$). In this case **Y** and **X** are connected by the relation of the inclusion.

An ordered tuple of named sets $\Theta = [\mathbf{X}_1, \mathbf{X}_2, ..., \mathbf{X}_k]$ where for all i=1, ..., k-1 the condition $N(\mathbf{X}_i) \cap S(\mathbf{X}_{i+1}) \neq \emptyset$ is fulfilled is called *chain of named sets*.

The number k is called a length of the chain Θ.

A tuple of named sets $\Xi_1 = [\mathbf{X}, \mathbf{Y}_1, \mathbf{Y}_2, ..., \mathbf{Y}_n]$ where for all i=1,...,n the condition $N(\mathbf{Y}_i) \cap S(\mathbf{X}) \neq \emptyset$ is fulfilled is called *one level hierarchy of named sets*.

If $N(\mathbf{Y}_i) \cap N(\mathbf{Y}_j) = \emptyset$ and $N(\mathbf{Y}_i) \subseteq S(\mathbf{X})$ for all i=1,...,n, j=1,...,n than Ξ is a *strong one level hierarchy of named sets*.

A tuple of named sets $\Xi_2 = [\mathbf{X}, \Xi_{1,1}, \Xi_{1,2}, ..., \Xi_{1,m}]$ where *sub-hierarhyies* $\Xi_{1j} = [\mathbf{Y}_j, \mathbf{Z}_1, \mathbf{Z}_2, ..., \mathbf{Z}_k]$, j=1,...,m are one level hierarchy of named sets is called *second level hierarchy of named sets*.

If Ξ_{1j}, j=1,...,m, are strong one level hierarchyies of named sets than Ξ_2 is a *strong second level hierarchy of named sets*.

A tuple of named sets $\Xi_n = [\mathbf{X}, \Xi_{n-1,1}, \Xi_{n-1,2}, ..., \Xi_{n-1,l}]$ where $\Xi_{n-1,i}$, i=1,...,l are n-1 level hierarchyies of named sets than Ξ_n is a *n-th level hierarchy of named sets*..

If all sub-hierarhyies of Ξ_n are strong hierarchyies of named sets than Ξ_n is a *strong n-th level hierarchy of named sets*.

Multi-domain information model (MDIM)

We will use strong hierarchies of named sets to create a specialized mathematical model for new kind of organization of information bases. The “Information Spaces” defined in the model are kind of strong hierarchies of enumerations (named sets).

The independence of dimensionality limitations is very important for developing new software systems aimed to process large volumes of high-dimensional data. To achieve this, we need information models and corresponding access methods to cross the boundary of the dimensional limitations and to obtain the possibility to work with large information spaces with variable and practically unlimited number of dimensions. A step in developing such methods is the **Multi-domain Information Model (MDIM)** introduced in [Markov, 1984; Markov, 2004]. Below we remember its main structures and operations.

Basic structures of MDIM

Main structures of MDIM are *basic information elements, information spaces, indexes* and *meta-indexes,* and *aggregates*. The definitions of these structures are given below:

- **Basic information elements**

The basic information element (*BIE*) of MDIM is an arbitrary long string of machine codes (bytes). When it is necessary, the string may be parceled out by lines. The length of the lines may be variable.

- **Information spaces**

Let the universal set ***UBIE*** be the set of all *BIE*.

Let $\boldsymbol{E}_1$ be a set of basic information elements. Let μ_1 be a function, which defines a biunique correspondence between elements of the set $\boldsymbol{E}_1$ and elements of the set $\boldsymbol{C}_1$ of positive integer numbers, i.e.:

$$\boldsymbol{E}_1 = \{e_i \mid e_i \in \boldsymbol{UBIE}, \ i=1,\dots, m_1\}.$$

$$\boldsymbol{C}_1 = \{c_1 \mid c_i \in N, \ i=1,\dots,m_1\}$$

$$\mu_1 \ E_1 \leftrightarrow C_1$$

The elements of $\boldsymbol{C}_1$ are said to be numbers (co-ordinates) of the elements of E_1.

The triple $\boldsymbol{S_1 = (E_1, \mu_1, C_1)}$ is said to be a ***numbered information space of level 1*** (one-dimensional or one-domain information space).

The triple $\boldsymbol{S_2 = (E_2, \mu_2, C_2)}$ is said to be a ***numbered information space of level 2*** (two-dimensional or multi-domain information space of level two) iff the elements of $\boldsymbol{E}_2$ are numbered information spaces of level one (i.e. belong to the set $\boldsymbol{NIS}_1$) and μ_2 is a function which defines a biunique correspondence between elements of $\boldsymbol{E}_2$ and elements of the set C_2 of positive integer numbers, i.e.:

$$\boldsymbol{E}_2 = \{e_i \mid e_i \in \boldsymbol{NIS_1}, \ i=1,\dots, m_2\}.$$

$$C_2 = \{c_i \mid c_i \in N, \ i=1,\dots,m_2\}$$

$$\mu_2 : E_2 \leftrightarrow C_2$$

The triple $\boldsymbol{S_n = (E_n, \mu_n, C_n)}$ is said to be a ***numbered information space of level n*** (n-dimensional or multi-domain information space) iff the elements of $\boldsymbol{E}_n$ are numbered information spaces of level n-1 (set $\boldsymbol{NIS}_{n-1}$) and μ_n is a function which defines a biunique correspondence between elements of $\boldsymbol{E}_n$ and elements of the set C_n of positive integer numbers, i.e.:

$$\boldsymbol{E}_n = \{e_j \mid e_j \in \boldsymbol{NIS_{n-1}}, \ j=1,\dots, m_n\}.$$

$$C_n = \{c_j \mid c_j \in N, \ j=1,\dots,m_n\}$$

$$\mu_n : E_n \leftrightarrow C_n$$

Every basic information element "e" is considered as an ***information space*** $\boldsymbol{S_0}$ of level 0. It is clear that the information space $\boldsymbol{S_0 = (E_0, \mu_0, C_0)}$ is constructed in the same manner as all others:

- The machine codes (bytes) b_i, i=1,…,m_0 are considered as elements of $\boldsymbol{E_0}$;
- The position p_i (natural number) of b_i in the string *e* is considered as co-ordinate of b_i, i.e.

$$C_0 = \{p_k \mid p_k \in N, k=1,\dots,m_0\},$$

- Function μ_0 is defined by the physical order of b_i in *e* and we have $\mu_0 : E_0 \leftrightarrow C_0$.

This way, the string $\boldsymbol{S_0}$ may be considered as a set of ***sub-elements (sub-strings).*** The number and length of the sub-elements may be variable. This option is very helpful but it closely depends on the concrete realizations and it is not considered as a standard characteristic of MDIM.

The information space $\boldsymbol{S_n}$, which contains all information spaces of a given application is called ***information base*** of level **n**. The concept information base without indication of the level is used as generalized concept to denote all available information spaces. For instance every relation data base may be represented as an ***information base of level 3*** which contains set of two dimensional tables.

- **Indexes and meta-indexes**

The sequence $A = (c_n, c_{n-1},\dots,c_1)$, where $c_i \in C_i$, i=1, …, *n* is called ***multidimensional space address*** of level ***n*** of a basic information element. Every space address of level ***m***, ***m < n***, may be extended to space address of level *n* by adding leading *n-m* zero codes. Every sequence of space addresses A_1, A_2, …, A_k**,** where ***k*** is arbitrary positive number, is said to be a ***space index***.

Every index may be considered as a basic information element, i.e. as a string, and may be stored in a point of any information space. In such case, it will have a multidimensional space address, which may be pointed in the other indexes, and, this way, we may build a hierarchy of indexes. Therefore, every index, which points only to indexes, is called ***meta-index***.

The approach of representing the interconnections between elements of the information spaces using (hierarchies) of meta-indexes is called ***poly-indexation.***

- **Aggregates**

Let $G = \{S_i \mid i=1,\dots,n\}$ be a set of numbered information spaces.

Let $\boldsymbol{\tau} = \{v_{ij} : S_i \longrightarrow S_j \mid i=const, j=1,\dots,n\}$ be a set of mappings of one "main" numbered information space $S_i \in G \mid i=const$, into the others $S_J \in G$, j=1, …, n , and, in particular, into itself.

The couple: $D = (G, \boldsymbol{\tau})$ is said to be an "***aggregate***".

It is clear, we can build **m** aggregates using the set G because every information space $S_J \in G$, j=1, …, n, may be chosen to be a main information space.

Operations in the MDIM

After defining the information structures, we need to present the operations, which are admissible in the model.

In MDIM, we assume that **all** information elements of **all** information spaces **exist**.

If for any S_i : $E_i = Ø \wedge C_i = Ø$, than it is called ***empty***.

Usually, most of the information elements and spaces are empty. This is very important for practical realizations.

Operations with basic information elements

Because of the rule that all structures exist, we need only two operations with a BIE:

- Updating;
- Getting the value.

For both operations, we need two service operations:

- Getting the length of a BIE;
- Positioning in a BIE.

Updating, or simply – ***writing*** the element, has several modifications with obvious meaning:

- Writing as a whole;
- Appending/inserting;
- Cutting/replacing a part;
- Deleting.

There is only one operation for getting the value of a BIE, i.e. ***read*** a portion from a BIE starting from given position. We may receive the whole BIE if the starting position is the beginning of BIE and the length of the portion is equal to the BIE length.

Operations with spaces

We have only one operation with a **single space** – *clearing (deleting) the space*, i.e. replacing all BIE of the space with Ø (empty BIE). After this operation, all BIE of the space will have zero length. Really, the space is cleared via replacing it with empty space.

We may provide two operations with **two spaces**: (1) *copying* and (2) *moving* the first space in the second. The modifications concern how the BIE in the recipient space are processed. We may have:

- Copy/move with clearing the recipient space;
- Copy/move with merging the spaces.

The first modifications first clear the recipient space and after that provide a copy or move operation.

The second modifications may have two types of processing: destructive or constructive. The ***destructive merging*** may be "conservative" or "alternative". In the conservative approach, the BIE of recipient space remains in the result if it is with none zero length. In the other approach – the BIE from donor space remains in the result. In the ***constructive merging*** the result is any composition of the corresponding BIE of the two spaces.

Of course, the move operation deletes the donor space after the operation.

Special kind of operations concerns the navigation in a space. We may receive the space address of the **next** or **previous, empty** or **non-empty** elements of the space starting from any given co-ordinates.

The possibility to count the number of non empty elements of a given space is useful for practical realizations.

- **Operations with indexes, meta-indexes and aggregates**

Operations with indexes, meta-indexes, and aggregates in the MDIM are based on the classical logical operations – intersection, union, and supplement, but these operations are not so trivial. Because of the complexity of the structure of the information spaces, these operations have two different realizations.

Every information space is built by two sets: the set of co-ordinates and the set of information elements. Because of this, the operations with indexes, meta-indexes, and aggregates may be classified in two main types:

- Operations based only on co-ordinates, regardless of the content of the structures;
- Operations, which take in account the content of the structures.

The operations based only on the co-ordinates are aimed to support information processing of analytically given information structures. For instance, such structure is the table, which may be represented by an aggregate. Aggregates may be assumed as an extension of the relations in the sense of the model of Codd [Codd, 1970]. The relation may be represented by an aggregate if the aggregation mapping is one-one mapping. Therefore, the aggregate is a more universal structure than the relation and the operations with aggregates include those of relation theory. What is the new is that the mappings of aggregates may be not one-one mappings.

In the second case, the existence and the content of non empty structures determine the operations, which can be grouped corresponding to the main information structures: elements, spaces, indexes, and meta-indexes. For instance, such operation is the **projection**, which is the analytically given space index of non-empty structures. The projection is given when some coordinates (in arbitrary positions) are fixed and the other coordinates vary for all possible values of coordinates, where non-empty elements exist. Some given values of coordinates may be omitted during processing.

Other operations are transferring from one structure to another, information search, sorting, making reports, generalization, clustering, classification, etc.

Hashing

A *set abstract data type* (set ADT) is an abstract data type that maintains a set ***S*** under the following three operations:

1. *Insert(x)*: Add the key *x* to the set.
2. *Delete(x)*: Remove the key *x* from the set.
3. *Search(x)*: Determine if *x* is contained in the set, and if so, return a pointer to *x*.

One of the most practical and widely used methods of implementing the set ADT is with *hash tables* [Morin, 2005].

The simplest implementation of such data structure is an ordinary array, where *k*-th element corresponds to key *k*. Thus, we can execute all operations in O(1). It is impossible to use this implementation, if the total number of keys is large [Kolosovskiy, 2009].

The main idea behind all hash table implementations is to store a set of *n* = |***S***| elements in an array (the hash table) *A* of length m. In doing this, we require a function that maps any element *x* to an array location. This function is called a *hash function h* and the value *h(x)* is called the *hash value of x*. That is, the element *x* gets stored at the array location *A*[*h(x)*].

The occupancy of a hash table is the ratio α = *n/m* of stored elements to the length of *A* [Morin, 2005].

We have two cases: (1) m ≥ n and (2) m ≤ n:

- In the first case (m ≥ n) we may expect so called ***perfect hashing*** where every element may be stored in separate cell of the array. In other words, if we have a collection of n elements whose keys are unique integers in (1, m), where m ≥ n, then we can store the items in a direct address table, T[m], where T_i is either empty or contains one of the elements of our collection.

- In the second case (m ≤ n) we may expect so called "***collisions***" when two or more elements have to be stored in the same cell f the array.

If we work with two or more keys, which have the *same hash value*, these keys map to the same cell in the array. Such situations are called *collisions*. There are two basic ways to implement hash tables to resolve collisions:

- Chained hash table;
- Open-address hash table.

In ***chained hash table*** each cell of the array contains the linked list of elements, which have corresponding hash value. To add (delete, search) element in the set we add (delete, search) to corresponding linked list. Thus, time of execution depends on length of the linked lists.

In ***open-address hash table*** we store all elements in one array and resolve collisions by using other cells in this array. To perform insertion we examine some slots in the table, until we find an empty slot or understand that the key is contained in the table. To perform search we execute similar routine [Kolosovskiy, 2009].

The study of hash tables follows two very different lines:

1) Integer universe assumption;
2) Random probing assumption.

Integer universe assumption: All elements stored in the hash table come from the universe $U = \{0,...,u-1\}$. In this case, the goal is to design a hash function $h : U \rightarrow \{0, ..., m-1\}$ so that for each $I \in \{0,...,m-1\}$, the number of elements $x \in \boldsymbol{S}$ such that $h(x) = i$ is as small as possible. Ideally, the hash function h would be such that each element of $\boldsymbol{S}$ is mapped to a unique value in $\{0, ..., m-1\}$.

Historically, the **integer universe assumption** seems to have been justified by the fact that any data item in a computer is represented as a *sequence of bits that can be interpreted as a binary number*.

However, many complicated data items require a large (or variable) number of bits to represent and this make the size of the universe very large. In many applications *u* is much larger than the largest integer that can fit into a single word of computer memory. In this case, *the computations performed in number-theoretic hash functions become inefficient*. This motivates the second major line of research into hash tables, based on *Random probing assumption*.

Random probing assumption: Each element *x* that is inserted into a hash table is a black box that comes with an infinite random probe sequence $x_0, x_1, x_2, ...$ where each of the x_i is independently and uniformly distributed in {*0, ..., m−1*}.

Both the integer universe assumption and the random probing assumption have their place in practice.

When there is an easily computing mapping of data elements onto machine word sized integers then hash tables for integer universes are the method of choice.

When such a mapping is not so easy to compute (variable length strings are an example) it might be better to use the bits of the input items to build a good pseudorandom sequence and use this sequence as the probe sequence for some random probing data structure [Morin, 2005].

Perfect hash function

We consider hash tables under the *integer universe assumption*, in which the key values *x* come from the universe *U* = {0*, ..., u−1*}. A hash function *h* is a function whose domain is *U* and whose level is the set {*0, ..., m−1*}, $m \leq u$.

A hash function h is said to be a ***perfect hash function*** for a set $S \subseteq U$ if, ***for every x ∈ S, h(x) is unique.***

A *perfect hash function h* for *S* is ***minimal*** if *m* = |*S*|, i.e., *h* is a bisection between *S* and {0, ..., *m* − 1}. Obviously a minimal perfect hash function for *S* is desirable since it allows us to store all the elements of *S* in a single array of length *n*. Unfortunately, perfect hash functions are rare, even for *m* much larger than *n* [Morin, 2005].

The set of elements, *S*, may be:

- *Static* (no updates);
- *Dynamic* where fast queries, insertions, and deletions must be made on a large set.

*"**Dynamic perfect hashing**"* is useful for the second type of situations. In this method, the entries that hash to the same slot of the table are organized as separate *second-level hash table*. If there are *k* entries in this set *S*, the second-level table is allocated with k^2 slots, and its hash function is selected at random from a universal hash function set so that it is *collision-free* (i.e. a perfect hash function). Therefore, the look-up cost is guaranteed to be O(1) in the worst-case [Dietzfelbinger et al, 1994].

Perfect hashing can be used in many applications in which we want to assign a unique identifier to each key without storing any information on the key. One of the most obvious applications of perfect hashing (or k-perfect hashing) is when we have a small fast memory in which we can store the perfect hash function while the keys and associated satellite data are stored in slower but larger memory. The size of a block or a transfer unit may be chosen so that *k* data items can be retrieved in one read access. In this case we can ensure that data associated with a key can be retrieved in a single probe to slower memory. This has been used for example in hardware routers.

Perfect hashing has also been found to be competitive with traditional hashing in internal memory on standard computers. *Recently perfect hashing has been used* ***to accelerate algorithms on graphs*** *when the graph representation does not fit in main memory* [Belazzougui et al, 2009].

For the purposes of CRP we need possibility to use *perfect hashing with dynamic and very large (practically – unlimited) set, S, of elements with variable length of strings*. In this case, the computing mapping of data elements onto machine word sized integers is not so easy to compute (we have long strings with variable length). In the same time, we could not use the bits of the input items to build a good pseudorandom sequence and use this sequence as the probe sequence for some random probing data structure, because of very large, unlimited, set, *S*, of elements.

Tries

> *“As defined by me, nearly 50 years ago, it is properly pronounced "tree" as in the word "retrieval". At least that was my intent when I gave it the name "Trie". The idea behind the name was to combine reference to both the structure (a tree structure) and a major purpose (data storage and retrieval)”.*
>
> *Edward Fredkin, July 31, 2008*

Trie is a tree for storing strings in which there is one node for every common prefix. The strings are stored in extra leaf nodes.

A *trie* can be thought of as an *m*-ary tree, where *m* is the number of characters in the alphabet. A search is performed by examining the key one character at a time and using an *m*-way branch to follow the appropriate path in the trie, starting at the root. In other words, in the *multi-way trie* (Figure 1), each node has a potential child for each letter in the alphabet. Below is an example of a multi-way trie indexing the three words BE, BED, and BACCALAUREATE [Pfenning, 2012].

Tries are distinct from the other data structures because they explicitly assume that the keys are a sequence of values over some (finite) alphabet, rather than a single indivisible entity. Thus tries are particularly well-suited for handling variable-length keys. Also, when appropriately implemented, tries can provide compression of the set represented, because common prefixes of words are combined together; words with the same prefix follow the same search path in the trie [Sahni, 2005].

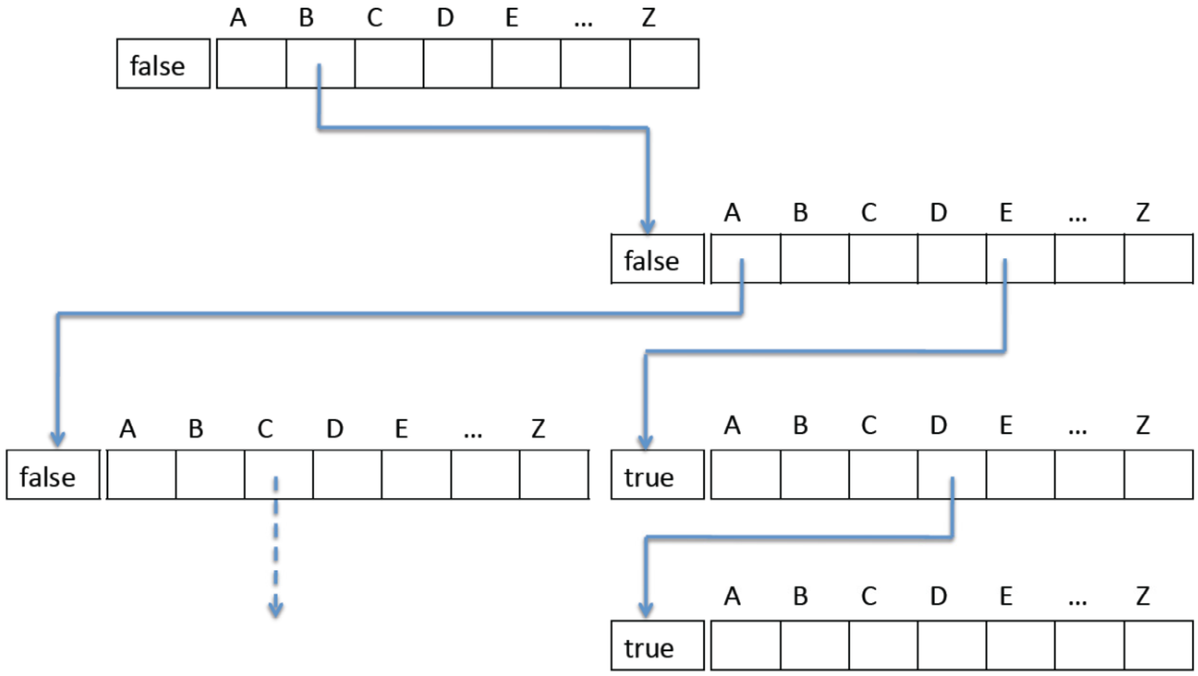


Figure 1. Example of multi-way trie [Pfenning, 2012]

Burst Tries

The tree data structures compared to hashing have three sources of inefficiency [Heinz et al, 2002]:

- First, the average search lengths is surprisingly high, typically exceeding ten pointer traversals and string comparisons even on moderate-sized data sets with highly skew distributions. In contrast, a search under hashing rarely requires more than a string traversal to compute a hash value and a single successful comparison;
- Second, for structures based on Binary Search Trees (BSTs), the string comparisons involved redundant character inspections, and were thus unnecessarily expensive. For example, given the query string “middle” and given that, during search, “Michael” and “midfield” have been encountered, it is clear that all subsequent strings inspected must begin with the prefix “mi”;
- Third, in tries the set of strings in a subtrie tends to have a highly skew distribution: typically the vast majority of accesses to a subtrie are to find one particular string. Thus use of a highly time-efficient, space-intensive structure for the remaining strings is not a good use of resources [Heinz et al, 2002].

These considerations led to the burst trie. A ***burst trie*** is an *in-memory* data structure, designed for sets of records that each has a unique string that identifies the record and acts as a key. Formally, a string ***s*** with length *n* consists of a series of symbols or characters c_i for *i*=0;...;*n*, chosen from an alphabet A of size |A|. It is assumed that |A| is small, typically no greater than 256 [Heinz et al, 2002].

A ***burst trie*** consists of three distinct components (Figure 2): a set of records, a set of containers, and an access trie.

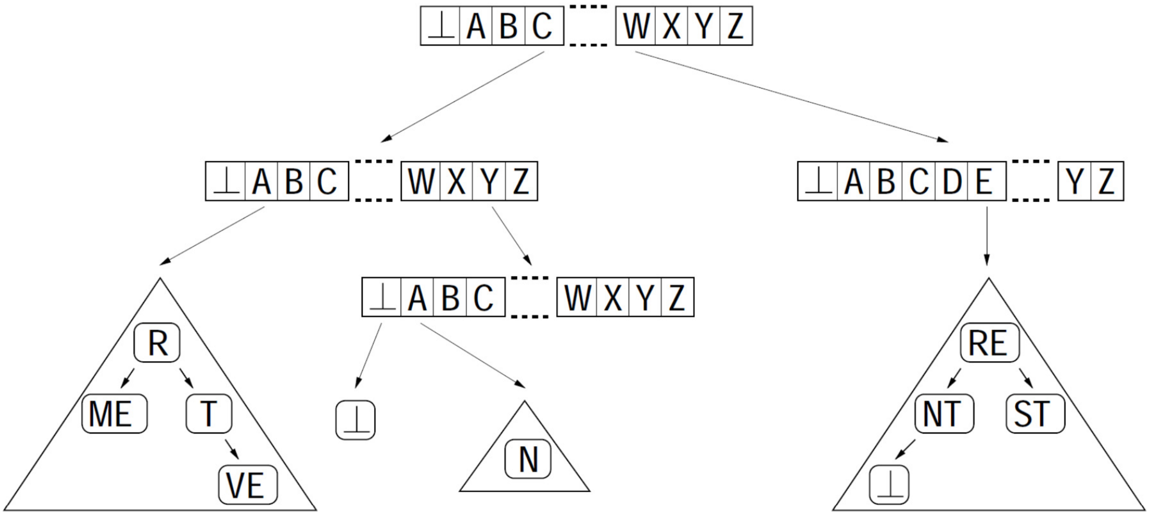


Figure 2. Burst trie with BSTs used in containers [Heinz et al, 2002]

Records. A record contains a string; information as required by the application using the burst trie (that is, for information such as statistics or word locations); and pointers as required to maintain the container holding the record. *Each string is unique*;

Containers. A container is a small set of records, maintained as a simple data structure such as a list or a *binary search tree* (BST). For a container at depth *k* in a burst trie, all strings have length at least k and the first k characters of all strings are identical. It is not necessary to store these first k characters. Each container also has a header, for storing the statistics used by heuristics for bursting. Thus a particular container at depth 3 containing “author” and “automated” could also contain “autopsy” but not “auger”;

Access trie. An access trie is a trie whose leaves are containers. Each node consists of an array *p*, of length |A|, of pointers, each of which may point to either a trie node or a container, and a single empty-string pointer to a record. The |A| array locations are indexed by the characters *c*∈A. The remaining pointer is indexed by the empty string.

The depth of the root is defined to be 1. Leaves are at varying depths.

A burst trie can be viewed as a generalization of other proposed variants of trie.

Figure 2 shows an example of a burst trie storing ten records whose keys are “came”, “car”, “cat”, “cave”, “cy”, “cyan”, “we”, “went”, “were”, and “west” respectively. In this example, the alphabet A is the set of letters from A to Z, and in addition an empty string symbol ⊥ is shown; the container structure used is a BST. In this figure, the access trie has four nodes, the deepest at depth 3. The leftmost container has four records, corresponding to the strings “came”, “car”, “cat”, and “cave”. One of the strings in the rightmost container is “⊥”, corresponding to the string “we”. The string “cy” is stored wholly within the access trie, as shown by the empty-string pointer to a record, indexed by the empty string [Heinz et al, 2002].

Natural Language Addressing

Analyzing Figure 1 and Figure 2, one may see a common structure in both figures. It is *a trie which leafs are containers*. In Figure 1 leafs are Social Security Numbers (SS#) and in Figure 2 leafs are Binary Search Trees (BST). In addition, Figure 1 looks as it is created from many connected Perfect Hash Tables (PHT).

In addition, if we take in account the possibilities of MDIM, we may use for realization a multi-way burst trie which:

- Nodes are PHT with entries for all numbers of given interval, for instance (0, 2^{32}-1);
- Containers may hold subordinated burst tries.

One very important consequence is to use as interval only the numbers which are codes of letters in any encoding system: ASCII, UNICODE16, or UNICODE32. This case is called “Natural Language Addressing” (NL-addressing) [Ivanova et al, 2013; Ivanova, 2014a; Ivanova, 2014b].

The idea of NL-addressing is to use encoding of the name both as relative address and as route in a Multi-dimensional information space and this way to speed the access to stored information. For instance, let have the next definition: “Pirrin: A mountain with co-ordinates (x, y)”. In the computer memory, it may be stored in a file at relative address “50067328” and the corresponded index couple may be: (“Pirrin”, “50067328”). At the memory address “50067328” the main text, “A mountain ... (x,y)” will be stored. To read/write the main text, firstly we need to find name “Pirrin” in the index and after that to access memory address “50067328” to read/write the definition. If we assume that name “Pirrin” in the computer memory is encoded by six numbers (letter codes), for instance by using ASCII encoding system Pirrin is encoded as (80, 105, 114, 114, 105, 110), than we may use these codes for direct address to memory, i.e. (“Pirrin”, “80, 105, 114, 114, 105, 110”).

Above we have written two times the same name as letters and codes. Because of this we may omit this couple and index, and read/write directly to the address “80, 105, 114, 114, 105, 110”. For human this address will be shown as “Pirrin”, but for the computer it will be “80, 105, 114, 114, 105, 110”.

Multi-domain access method “ArM32”

Perfect hash tables and burst tries give very good starting point. The main problem is that they are designed as *structures in the main memory* which has limited size, especially in small desktop and laptop computers.

For practical implementation aimed to store very large perfect hash tables and burst tries *in the external memory* (hard disks) we need relization in accordance to the real possibilities. One possible solution is to use “Multi-Domain Information Model” (MDIM) [Markov, 1984] and corresponded to it software tools.

During the last three decades, MDIM has been discussed in many publications. See for instance [Markov et al, 1990; Markov, 2004; Markov et al, 2013].

The the corresponded to MDIM access method and its different program realizations during the years have different names: **M**ulti-**D**omain **A**ccess **M**ethod (MDAM), **Ar**chive **M**anager (ArM), and **N**atural **L**anguage Addressing **Ar**chive **M**anager (NL-ArM), **Big** Data **Ar**chive **M**anager **(BigArM)** (Table 1).

Developing the method and all projects of its realizations had been done by Krassimir Markov.

The program realizations had been done by:

- Krassimir Markov (MDAM0, MDAM1, MDAM2, MDAM3);
- Dimitar Guelev (MDAM4);
- Todor Todorov (MDAM5 written on Assembler with interfaces to PASCAL and C, MDAM5 rewritten on C for IBM PC);
- Vasil Nikolov (MDAM5 interface for LISP, MDAM6);
- Vassil Vassilev (ArM7 and ArM8);
- Ilia Mitov and Krassimira Minkova Ivanova (ArM 32);
- Vitalii Velychko (ArM32 interface to Java);
- Krassimira Borislavova Ivanova (NL-ArM).

Table 1. Realizations of MDAM

no.	name	year	machine	type	language and	operating system
0	MDAM0	1975	MINSK 32	37 bit	Assembler	Tape OS
1	MDAM1	1981	IBM 360	32 bit	FORTRAN	DOS 360
2	MDAM2	1983	PDP 11	16 bit	FORTRAN	DOS 11
3	MDAM3	1985	PDP 11	16 bit	Assembler	DOS 11
4	MDAM4	1985	Apple II	8 bit	UCSD Pascal	Disquette OS
5	MDAM5	1986	IBM PC	16 bit	Assembler, C	MS DOS
6	MDAM6	1988	SUN	32 bit	C	UNIX
7	ArM7	1993	IBM PC	16 bit	Assembler	MS DOS 3
8	ArM8	1998	IBM PC	16 bit	Object Pascal	MS Windows 16 bit
9	ArM32	2003	IBM PC	32 bit	Object Pascal	MS Windows 32 bit
10	NL-ArM	2012	IBM PC	32 bit	Object Pascal	MS Windows 32 bit
11	BigArM	2015 ... under developing		64 bit	Pascal, C, Java	MS Windows, Linux, Cloud

For a long period, MDIM has been used as a basis for organization of various information bases.

One of the first goals of the development of MDIM was representing the digitalized military defense situation, which is characterized by a variety of complex objects and events, which occur in the space and time and have a long period of variable existence [Markov, 1984]. The great number of layers, aspects, and interconnections of the real situation may be represented only by information spaces' hierarchy. In addition, the different types of users with individual access rights and needs insist on the realization of a special tool for organizing such information base.

Over the years, the efficiency of MDIM is proved in wide areas of information service of enterprise managements and accounting. For instance, the using MDIM permits omitting the heavy work of creating of OLAP structures [Markov, 2005].

ArM32

Crrent realization of MDIM, respectively – MDAM, is the Archive Manager – "ArM32" developed for MS Windows (32 bit) [Markov, 2004; Markov et al, 2008] and its upgrate to NL-ArM.

The ArM32 elements are organized in numbered information spaces with variable levels. There is no limit for the levels of the spaces. Every element may be accessed by a corresponding multidimensional space address (coordinates) given via coordinate array of type cardinal. At the first place of this array, the space level needs to be given. Therefore, we have two main constructs of the physical organizations of ArM32 information bases – numbered information spaces and elements.

The ArM32 Information space (IS) is realized as a (*perfect*) *hash table stored in the external memory.* Every IS has 2^{32} entries (elements) numbered from 0 up to 2^{32}-1. The number of the entry (element) is called its *co-ordinate*, i.e. the co-ordinate is a 32 bit integer value and it is the number of the entry (element) in the IS.

Every entry is connected to a container with variable length from zero up to 1G bytes. If the container holds zero bytes it is called "empty". In other words, in ArM32, the length of the element (string) in the container may vary from 0 up to 1G bytes. There is no limit for the number of containers in an archive but their total length plus internal indexes could not exceed 2^{32} bytes in a single file.

If all containers of an IS hold other IS, it is called "*IS of corresponded level"* depending of the depth of including subordinated IS. If containers of given IS hold arbitrary information but not other IS, it is called "*Terminal IS*".

To locate a container, one has to define ***the path in hierarchy*** using a ***co-ordinate array*** with all numbers of containers starting from the one of the *root* information space up to the *terminal* information space which is owner of the container.

The hierarchy of information spaces may be not balanced. In other words, it is possible to have branches of the hierarchy which have different depth.

*In ArM32, we assume that **all possible** information spaces **exist**.*

If all containers of the information space are empty, it is called "***empty***".

Usually, most of the ArM32 information spaces and containers are empty. “Empty” means that corresponded structure (space or container) does not occupy disk space. This is very important for practical realizations.

Remembering that **Trie** is a tree for storing strings in which there is one node for every common prefix and the strings are stored in extra leaf nodes, we may say the ArM32 has analogous organization and *can be used to store (burst) tries*.

- **Functions of ArM32**

ArM32 is realized as set of functions wich may be executed from any user program. Because of the rule that all structures of MDIM exist, we need only two main functions with containers (elements):

- Get the value of a container (as whole or partially);
- Update a container (with several variations).

Because of this, the main ArM32 functions with information elements are:

- *ArmRead* (reading a part or a whole element);
- *ArmWrite* (writing a part or a whole element);
- *ArmAppend* (appending a string to an element);
- *ArmInsert* (inserting a string into an element);
- *ArmCut* (removing a part of an element);
- *ArmReplace* (replacing a part of an element);
- *ArmDelete* (deleting an element);
- *ArmLength* (returns the length of the element in bytes).

MDIM operations with information spaces are over:

- **Single space** – *clearing the space*, i.e. updating all its containers to be empty;
- **Two spaces** – there exist several such type of operations. The most used is copying of one space in another, i.e. copying the contents of containers of the first space in the containers of the second. Moving and comparing operations are available, too.

The corresponded ArM32 functions over the spaces are:

- *ArmDelSpace* (deleting the space);
- *ArmCopySpace* and *ArmMoveSpace* (copying/moving the firstspace in the second in the frame of one file);
- *ArmExportSpace* (copying one space from one file to the other space, which is located in another file).

The ArM32 functions, aimed to serve the navigation in the information spaces return the space address of the **next** or **previous, empty** or **non-empty** elements of the space starting from any given co-ordinates. They are *ArmNextPresent, ArmPrevPresent, ArmNextEmpty*, and *ArmPrevEmpty*.

The ArM32 function, which create indexes, is *ArmSpaceIndex* – returns the space index of the non-empty structures in the given information space.

The service function for counting non-empty ArM32 elements or subspaces is *ArmSpaceCount* – returns the number of the non-empty structures in given information space.

ArM32 engine supports multithreaded concurrent access to the information base in real time. Very important characteristic of ArM32 is possibility not to occupy disk space for empty structures (elements or spaces). Really, only non-empty structures need to be saved on external memory.

Summarizing, the advantages of the ArM32 are:

- Possibility to build growing space hierarchies of information elements;
- Great power for building interconnections between information elements stored in the information base;
- Practically unlimited number of dimensions (this is the main advantage of the numbered information spaces for well-structured tasks, where it is possible "***to address, not to search***").

NL-ArM access method

MDAM and respectively ArM32 are not ready to support NL-addressing. We have to upgrade them for ensuring the features of NL-addressing. The new access method is called **NL-ArM** (Natural Language Addressing Archive Manager).

The program realization of NL-ArM is based on a specialized hash function and two main functions for supporting the NL-addressing access.

In addition, several operations were realized to serve the work with thesauruses and ontologies as well as work with graphs.

➢ NL-ArM hash function

The NL-ArM hash function is called "*NLArmStr2Addr*". It converts a string to space path. Its algorithm is simple: four ASCII symbols or two UNICODE 16 symbols form one 32 bit co-ordinate word. This reduces the space' level four, respectively – two, times. The string is extended with leading zeroes if it is needed. UNICODE 32 does not need converting – one such symbol is one co-ordinate word.

There exists a reverse function, "*NLArmAddr2Str*". It converts space address in ASCII or UNICODE string. The leading zeroes are not included in the string.

The functions for converting are not needed for the end-user because they are used by the NL-ArM upper level operations given below.

All NL-ArM operations access the information by NL-addresses (given by a NL-words or phrases). Because of this we will not point specially this feature.

➢ NL-ArM operations with terminal containers

Terminal containers are those which belong to terminal information spaces. They hold strings up to 1GB long.

There are two main operations with strings of terminal containers:

- *NLArmRead* – read from a container (all string or substring);
- *NLArmWrite* – update the container (all string or substring).

Additional operations are:

- *NLArmAppend* (appending a substring to string of the container);
- *NLArmInsert* (inserting a substring into string of the container);
- *NLArmCut* (removing a substring from the string of the container);
- *NLArmReplace* (replacing a substring from the string of the container);
- *NLArmDelete* (empting the container);
- *NLArmLength* (returns the length of the string in the container in bytes).

In general, the container may be assumed not only as up to 1GB long string of characters but as some other information again up to 1GB. As a rule, the access methods do not interpret the information which is transferred to and from the main memory. It is important to have possibility to access information in the container as a whole or as set of concatenated parts.

Assuming that all containers exist but some of them are empty, we need only two main operations:

1) To update (write) the string or some of its parts.
2) To receive (read) the string or some of its parts.

The additional operations are modifications of the classical operations with strings applied to this case.

To access information from given container, NL-ArM needs the path to this container and buffer from or to which the whole or a part of its content will be transferred. Additional parameters are length in bytes and possibly - the starting position of substring into the string. When string has to be transferred as a whole, the parameters are the length of the string and zero as number of the starting position.

- **NL-ArM operations with information spaces (hash tables)**

With information spaces we may provide service operations with hash tables such as counting empty or non-empty containers, copying or moving strings of substrings from containers one to those of another terminal information space. We will not use these operations in the frame of this work.

Requirements to BigArM realization characteristics

Main characteristics of program realizations of MDAM are shown in Table 2.

Using ArM32 engine we have great limit for the number of dimensions as well as for the number of elements on given dimension. The boundary of this limit in the current realization of ArM32 engine is 2^{32} for every dimension as well as for number of dimensions. Of course, another limitation is the maximum length of the files, which depends on the possibilities of the operating systems and realization of ArM. Main limitation of ArM32 is that the length of archive files may be 4GB long. This cause that in practical implementations we have not so big number of dimensions (usually it is about 200).

What is needed is to extend possibilities of ArM32 from 32 bit up to 64 bit addressing capabilities and to rationalize the internal hash structures to speed access from milliseconds down to microseconds per one access operation. This will be done in ongoing developing of its new version called “BigArM” for 64 bit machines and operating systems like MS Windows and Linux. In addition, BigArM will permit new kind of Cloud processing of Big Data, called “Collect/Report Paradigm” (CRP) [Markov et al, 2014; Markov & Ivanova, 2015].

Table 2. Main characteristics of program realizations of MDAM

№	name	max dimensions	max size of element	max number of elements	max size of archive	max size of information base	access time
0	MDAM0	1	128 bytes	128	512 words	16K words	minutes
1	MDAM1	1	256 bytes	256	1 KB	10 MB	seconds
2	MDAM2	2	256 bytes	2^{31} (10 000)	32 KB	4 MB	seconds
3	MDAM3	2	256 bytes	2^{31} (10 000)	32 KB	4 MB	seconds
4	MDAM4	1	80 bytes	25	30 elements	4KB	deciseconds
5	MDAM5	2	64 KB	2^{31} (1 000 000)	32KB	80 MB	centiseconds
6	MDAM6	2	64 KB	2^{31} (1 000 000)	32KB	90 MB	centiseconds
7	ArM7	2+2	1 GB	2^{60}	4 GB	10 GB	milliseconds
8	ArM8	2+2	1 GB	2^{60}	4 GB	10 GB	milliseconds
9	ArM32	200	1 GB	2^{64} (max 4G)	4 GB	1TB	milliseconds
10	NL-ArM	200	1 GB	2^{64} (max 4G)	4 GB	1TB	milliseconds
11	BigArM	2^{32}	4 GB	2^{64}	1 PB	1 YB	microseconds

Conclusion

In this survey we presented mathematical and informational foundations as well as requirements to realization characteristics BigArM - an access method for storing and accessing Big Data. It is under development. Firstly, we outlined the needed basic mathematical concepts, the Names Sets, and hierarchies of named sets aimed to create a specialized model for organization of information bases called “Multi-Domain Information Model” (MDIM). The “Information Spaces” defined in the model are kind of strong hierarchies of enumerations (named sets). Further we remembered the main features of hashing and types of hash tables as well as the idea of “Dynamic perfect hashing” and “Trie”, especially – the “Burst trie”. Hash tables and tries give very good starting point. The main problem is that they are designed as structures in the main memory which has limited size, especially in small desktop and laptop computers. To solve this problem, dynamic perfect hashing and burst tries will be realized as external memory structures in BigArM.

Special attention we have paid to MDIM and its realizations ArM2 and NL-ArM. The program realization of NL-ArM is based on specialized hash functions and two main functions for supporting the NL-addressing access. In addition, several operations were realized to serve the work with thesauruses and ontologies as well as work with graphs.

Finaly, we have presented the main requirements to BigArM realization characteristics. The expected project characteristics of BigArM are sumarized in Table 3.

Table 3. Project characteristics of BigArM

Programming language	Object Pascal, C, Java
Operational environment	Windows, Linux, Cloud
Maximal size of the elements in the archive	**4 GB**
Maximal size of the archive	**2^{64} (>1 PB = 2^{50})**
Maximal sizeof the information base	**no limit (>1 YB = 2^{80})**
Access time	**microseconds**
Dimensions of the information spaces	**variable up to max 2^{32}**
Number of elements in an archive	**2^{64}**
Main technologies for accessing data	– **Direct R/W addressing** – **NL R/W addressing** – **Collect/Report paradigm**

Bibliography

[Belazzougui et al, 2009] Djamal Belazzougui, Fabiano C. Botelho, Martin Dietzfelbinger, “Hash, Displace, and Compress”, In: Algorithms - ESA 2009 - 17th Annual European Symposium, Copenhagen, Denmark, September 7-9, 2009, Proceedings. Lecture Notes in Computer Science Volume 5757, Springer, 2009, pp 682-693. DOI 10.1007/978-3-642-04128-0_61 Print ISBN: 978-3-642-04127-3 Online ISBN: 978-3-642-04128-0. http://link.springer.com/chapter/10.1007%2F978-3-642-04128-0_61 (accessed: 20.07.2013).

[Bourbaki, 1960] Bourbaki, N., “Theorie des Ensembles”, Hermann, Paris, 1960, English version: Bourbaki, N. Theory of Sets, Volume package: Elements of Mathematics. Springer, 1st ed. 1968, 2nd printing 2004, ISBN 978-3-540-22525-6, 414 p.

[Burgin & Gladun, 1989] Mark Burgin, Victor Gladun, “Mathematical Foundations of Semantic Networks Theory”, In: LNCS No.: 364, Springer, 1989. pp. 117-135.

[Burgin, 2010] Mark Burgin, “Theory of Information - Fundamentality, Diversity and Unification”, World Scientific Publishing Co. Pte. Ltd. Singapore, 2010, ISBN-13 978-981-283-548-2, 672 p.

[Burgin, 2011] Mark Burgin, “Theory of Named Sets”, Nova Science Publishers Inc (United States), 2011, ISBN-13: 9781611227888, 681 p.

[Codd, 1970] Codd, E., “A relation model of data for large shared data banks”, Magazine Communications of the ACM, 13/6, 1970, pp. 377-387

[Dietzfelbinger et al, 1994] Martin Dietzfelbinger, Anna Karlin, Kurt Mehlhorn, Friedhelm Meyer auf der Heide, Hans Rohnert, and Robert E. Tarjan, “Dynamic Perfect Hashing: Upper and Lower Bounds”, SIAM J. Comput, 23, 4, 1994, ISSN: 0097-5397, pp. 738-761, http://portal.acm.org/citation.cfm?id=182370# (accessed: 20.07.2013).

[Heinz et al, 2002] Steffen Heinz, Justin Zobel, Hugh E. Williams, “Burst Tries: A Fast, Efficient Data Structure for String Keys”, ACM Transactions on Information Systems (TOIS), Volume 20, Issue 2, April 2002, pp. 192 – 223, ACM New York, NY, USA, doi>10.1145/506309.506312, http://dl.acm.org/citation.cfm?id=506312 (accessed: 20.07.2013)

[Ivanova et al, 2013] Krassimira B. Ivanova, Koen Vanhoof, Krassimir Markov, Vitalii Velychko, “Introduction to the Natural Language Addressing”, International Journal "Information Technologies & Knowledge" Vol.7, Number 2, 2013, ISSN 1313-0455 (printed), 1313-048X (online), pp. 139–146.

[Ivanova, 2014a] Krassimira Ivanova, “Multi-Layer Knowledge Representation”, International Journal “Information Content and Processing”, Vol. 1, Number 4, 2014, ISSN 2367-5128 (printed), 2367-5152 (online), pp. 303 - 310.

[Ivanova, 2014b] Krassimira Ivanova, “Practical Aspects of Natural Language Addressing”, In: G. Setlak, K. Markov (ed.), Computational Models for Business and Engineering Domains, ITHEA®, 2014, Rzeszow, Poland, Sofia, Bulgaria, ISBN: 978-954-16-0066-5 (printed), ISBN: 978-954-16-0067-2 (online), pp. 172 – 186.

[Kolosovskiy, 2009] Kolosovskiy M., “Simple implementation of deletion from open-address hash table”, Cornell University Library, ArXiv e-prints, 2009, http://adsabs.harvard.edu/abs/2009arXiv0909.2547K (accessed: 20.07.2013)

[Markov & Ivanova, 2015] Markov Kr., Kr. Ivanova, “General Structure of Collect/Report Paradigm for Storing and Accessing Big Data”, Int. J. Information Theories and Applications, 22/3, 2015, pp. 266-290

[Markov et al, 1990] K. Markov, T. Todorov, V. Nikolov, “Multidomain Access Method for the IBM PC”, Research in Informatics, Vol. 3, Academie-Verlag Berlin, 1990, pp. 218-230.

[Markov et al, 2008] Markov, K., Ivanova, K., Mitov, I., & Karastanev, S., “Advance of the access methods”, International Journal of Information Technologies and Knowledge, 2(2), 2008, pp. 123–135.

[Markov et al, 2013] Markov, Krassimir, Koen Vanhoof, Iliya Mitov, Benoit Depaire, Krassimira Ivanova, Vitalii Velychko and Victor Gladun, "Intelligent Data Processing Based on Multi- Dimensional Numbered Memory Structures", Diagnostic Test Approaches to Machine Learning and Commonsense Reasoning Systems, IGI Global, 2013, pp. 156-184, doi:10.4018/978-1-4666-1900-5.ch007, ISBN: 978 1-4666-1900-5, EISBN: 978-1-4666- 1901-2
Reprinted in: Markov, Krassimir, Koen Vanhoof, Iliya Mitov, Benoit Depaire, Krassimira Ivanova, Vitalii Velychko and Victor Gladun, "Intelligent Data Processing Based on Multi-Dimensional Numbered Memory Structures", Data Mining: Concepts, Methodologies, Tools, and Applications, IGI Global, 2013, pp. 445-473, doi:10.4018/978-1-4666-2455-9.ch022, ISBN13: 978-1-4666-2455-9, EISBN13: 978-1-4666-2456-6

[Markov et al, 2014] Kr. Markov, Kr. Ivanova, K. Vanhoof, B. Depaire, V. Velychko, J. Castellanos, L. Aslanyan, St. Karastanev, “Storing Big Data Using Natural Language Addressing”, In: N. Lyutov (ed.), Int. Sc. Conference “Informatics in the Scientific Knowledge”, VFU, Varna, Bulgaria, 2014, ISSN: 1313-4345, pp. 147-164.

[Markov, 1984] Markov Kr., "A Multi-domain Access Method", Proceedings of the International Conference on Computer Based Scientific Research, Plovdiv, 1984, pp. 558-563.

[Markov, 2004] Markov, K., "Multi-domain information model", Int. J. Information Theories and Applications, 11/4, 2004, pp. 303-308

[Markov, 2005] Markov, K., "Building data warehouses using numbered multidimensional information spaces", International Journal of Information Theories and Applications, 12(2), 2005, pp. 193–199

[Morin, 2005] Pat Morin, "Hash tables", Chapter 9, of "Handbook of data structures and applications" /edited by Dinesh P. Mehta and Sartaj Sahni, Chapman & Hall/CRC computer & information science, ISBN 1-58488-435-5, 2005, 1321 p.

[Pfenning, 2012] Frank Pfenning, "Lecture Notes on Tries", Lecture 21, In 15-122: Principles of Imperative Computation November 8, 2012. http://www.cs.cmu.edu/~fp/courses/15122-f12/lectures/21-tries.pdf (accessed: 20.07.2013).

[Sahni, 2005] Sartaj Sahni, "Tries", Chapter 28, of "Handbook of data structures and applications" /edited by Dinesh P. Mehta and Sartaj Sahni, Chapman & Hall/CRC computer & information science, 2005, 1321 pages, ISBN 1-58488-435-5.

Authors' Information

***Krassimira Ivanova**– University of National and World Economy, Sofia, Bulgaria; Institute of Mathematics and Informatics, BAS, Bulgaria; e-mail: krasy78@mail.bg*

Major Fields of Scientific Research: Software Engineering, Business Informatics, Data Mining, Multidimensional multi-layer data structures in self-structured systems

TABLE OF CONTENTS