A Multimodal Biometric System Implemented within an Active Database Management System

Kornelije Rabuzin
University of Zagreb, Faculty of Organization and Informatics, Varaždin, Croatia
kornelije.rabuzin@foi.hr

Miroslav Bača and Mirko Maleković
University of Zagreb, Faculty of Organization and Informatics, Varaždin, Croatia
{miroslav.baca, mirko.malekovic}@foi.hr

Abstract— **Active databases have been used in many different fields to accomplish many different tasks. One of the main problems today that still have not been solved is authorization. In some situations it is enough that user knows e.g. password, in other situations it is enough that user posses e.g. a smart card, but, there are also situations where user's unique physical or psychological characteristics need to be measured. In this paper we are going to put accent on this third possibility. We will show how the concept of complex events presented in the active database theory could be used in order to build a multimodal biometric system. Especially, we will explore the paradigm of active rules and complex events, and apply them in order to implement a multimodal biometric system.**

***Index Terms*— Biometrics, Active databases, Triggers**

I. INTRODUCTION

Organization can be defined, as can be found in [15], as a set of people that are gathered in order to accomplish some common goal or goals that are of great importance for the organization itself. In order to fulfill these goals, people have to use certain resources (data, information, etc.). Some resources are restricted, i. e. known or accessed by only a small number of people. This is just one example why one has to be authorized in order to access some restricted resources. For example, if one wants to use e-mail, one has to have an account (a login name and a proper password), etc.

As it was already mentioned, a user could be authorized in three different ways (or their combinations): in some situations it is enough that user knows e.g. password, in other situations it is enough that user posses e.g. a smart card. But, there are situations where user's unique physical or psychological characteristics need to be measured.

When talking about passwords, people usually chose passwords that are easy, intuitive and not complex enough. Registration numbers or birth dates are used as well as names, passwords are usually to easy and written down; so in any case they are not a flawless solution. On the other hand, smart cards (or token) could be stolen which is not good either. In order to make a personal recognition, biometrics relies on who you are or what you do, as opposed to what you know (a password) or what you have (a card) [8]. Biometric features are intrinsic to every human and are therefore suitable to authorize particular user [8]. Biometric-based identification is preferred over traditional methods because a biometric cannot be forgotten or lost [20].

According to [17], database applications are mostly (still) passive i. e. they don't use active features (triggers) even though the underlying DBMS (DataBase Management System) may offer them. Because of that application performances are not so good and applications are harder to maintain. We will show later in the sample processing and decision-making that a generic biometric system is performed outside the database. The module responsible for comparison and authorisation is usually separated and because of that many biometric data must be extracted from the database. As a consequence we will show that a generic biometric system operates as a passive application; it means that decision-making is not performed within the database and that the performances of such an application could be (significantly) improved. That is why we will convert a (multimodal) biometric system into an active application i. e. we will use triggers; in that way the decision-making process will be performed within the database and active features are going to be used as well. In the end, the time needed for authorisation will be significantly reduced, as our preliminary results presented in the paper will show.

The rest of the paper is organized as follows: section 2 deals with biometrics, section 3 presents the theory of active databases, section 4 describes how we have modelled the authorisation (identification) problem using complex events and presents preliminary results, and finally, section 5 summarizes the paper.

II. BIOMETRICS

As we have already mentioned biometrics relies on who you are and how you behave; this was enough for the introduction, but now we will present some definitions that we have selected although many other, similar definitions can be found as well. Biometrics are automated methods of identifying a person or verifying the identity of a person based on a physiological or behavioural characteristic [4]. Biometrics is a method using physiological or behavioural features of a person for an automated detection and verification of their identity [10].

Many biometric characteristics are being used today, including fingerprint, DNA, iris pattern, retina, ear, face, thermogram, gait, hand geometry, palm-vein pattern, keystroke dynamics, smell, signature, and voice [10]. According to [1], the ideal biometric characteristic has to meet the following criteria: it has to be permanent and inalterable in terms of time, the procedure of gathering personal features has to be inconspicuous and conducted by means of devices involving minimum or no contact, it has to enable total automation of the system, and finally, the system has to be highly accurate and its operation speed such that it enables real-time operation. So, previously mentioned biometric characteristics that are being used could not be considered ideal, taking into account these criteria.

A generic model of a biometric system that consists of several important components can be seen in Fig 1. Basically, when the feature is extracted, data is compared against the stored template, and then the decision is made.

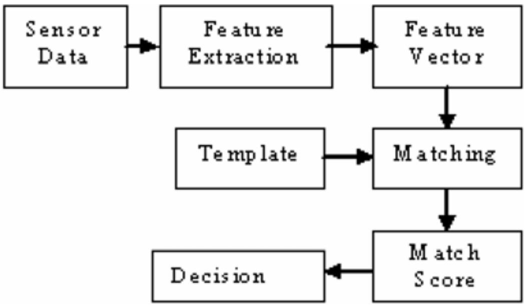

Figure 1. Biometric system [16]

In [20] biometric verification problem has been formulated as follows: "Let the stored biometric signal (template) of a person be represented as *S* and the acquired signal (input) for authentication be represented by *I*. Then the null and alternate hypotheses are:

H0: $I \neq S$, input fingerprint does not come from the same finger as the template,

H1: $I = S$, input fingerprint comes from the same finger as the template.

The associated decisions are as follows:

D0: person is an imposer,

D1: person is genuine.

The verification involves matching *S* and *I* using a similarity measure. If the matching score is less than some decision threshold *T*, then decide *D0*, else decide *D1*."

According to [21], Enrolment task creates association between identity and its biometric characteristics. In a Verification task enrolled user claims the identity and the system verifies the authenticity of the claim based on its biometric feature. And Identification task identifies enrolled user based on its biometric characteristics without user having to claim the identity. All the above can be seen in Fig. 2.

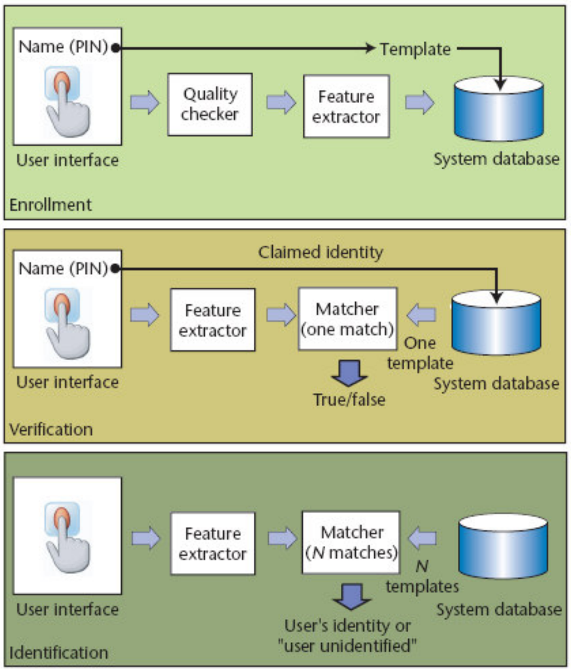

Figure 2. Block diagrams of Enrolment, Verification, and Identification tasks [21]

Another point of view is delineated in Fig. 3.:

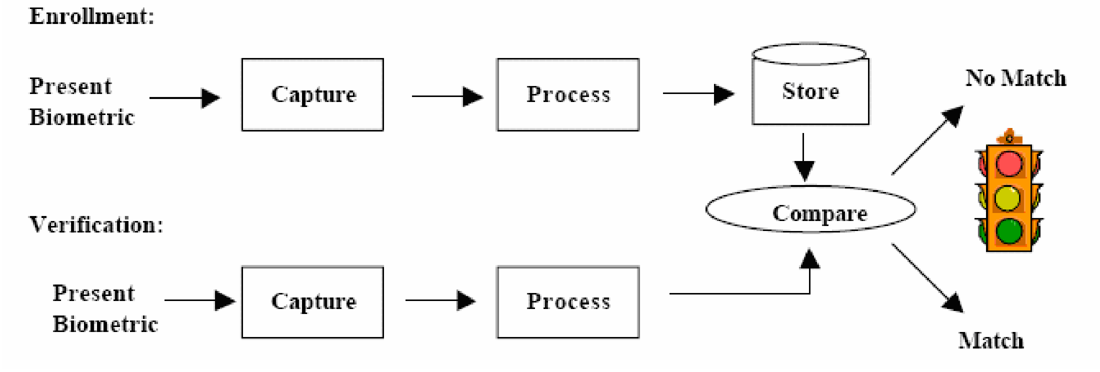

Figure 3. Enrolment and Verification [4]

According to [4], biometric authentication requires comparing registered or enrolled biometric sample (biometric template or identifier) against newly captured biometric sample (for example, a fingerprint captured during login process); identification is a much harder problem than verification because an identification system must perform a large number of comparisons.

Two basic types of biometric systems can be distinguished: unimodal and multimodal biometric system. Considering that neither of the biometric features is sufficiently reliable, single features can be combined together in one of two possible ways: unimodal or multimodal systems (that arises as an immediate solution) [8]. The main difference is that unimodal biometric system is based on just one biometric feature and it is typical to such approach that this (one) feature is singled out by means of several technologically distinct methods and systems [10]. On the other hand, multimodal biometric systems use several biometric features and technologies at the same time.

Although the second approach may seem more appropriate at first, several criteria are crucial that determine and influence the selection including the chief purpose of a system: the number and type of characteristics to be integrated, and so on. We have addressed these issues in a detail in [8] and [10].

For both types of systems (unimodal and multimodal) accuracy and speed are crucial [18]. The accuracy indicates the extent to which a multimodal biometric system is reliable and confidential when distinguishing a legitimate user from an imposter, while the speed of a multimodal biometric system indicates the time needed by the system to perform the personal identification [10].

We have already investigated some issues in the field of biometrics. In [12] we have explained how to use Walsh functions in order to pre-process fingerprints. In [14] we were discussing some security issues regarding palm recognition, and in [11] we have investigated how to build a database for fingerprints storage. Further on, in [13] we have tried to systemize different biometric techniques and methods that are being used, etc.

III. ACTIVE DATABASES

Active DataBase Management Systems (ADBMS) are database systems capable of reacting to some events of interest that can occur within the database, or outside database. The basic concept on which an ADBMS relies is the concept of ECA (Event Condition Action) or active rules. According to this concept, when certain events occur (ON EVENT), and some conditions are fulfilled (IF CONDITION), as a consequence some actions are performed automatically (THEN ACTION). Each ADBMS has a language that is used for trigger specification (definition), and possesses an execution model that determines how the rules are going to be executed.

An event can be defined as a state change of interest that requires intervention. Events can be divided into two categories: simple and complex events. Complex events are mostly based on simple ones, and simple events can be divided as follows (authors mostly agree on these basic types of events):

1. Database operations: INSERT, UPDATE, and/or DELETE,
2. Time events:
   a) *absolute* – a certain point of time,
   b) *periodic* - every day, month, etc., and
   c) *relative* - for example, 30 minutes after something else has happened,
3. Method events: method invocation,
4. Transaction events: for example BEGIN or COMMIT, and
5. Abstract events: some user defined events.

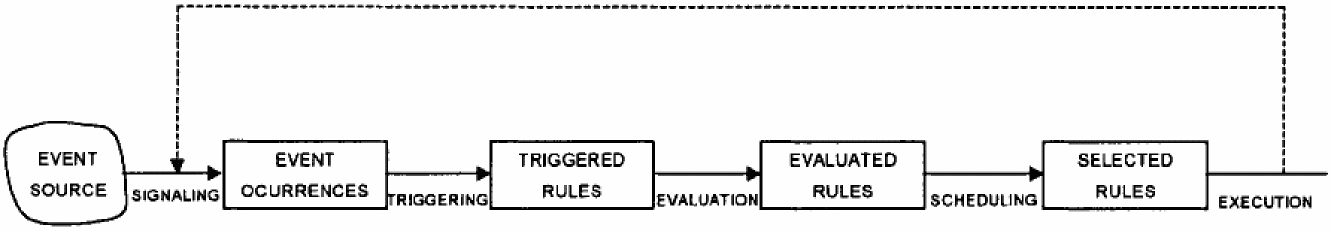

Figure 4. Rule execution [23]

Complex events could consist of one or more simple events connected with logical operators, but there are also some special kinds of complex events introduced during the years within many different projects (*REPEAT*, *NEGATION*, etc.). For example, if we had simple events E1 and E2, then E1∧E2 or E1∨E2 would represent a complex event.

As it has been already mentioned, when an event occurs the condition is evaluated, and then some actions are executed provided that condition evaluation was successful (Fig. 4). However, it is useful (sometimes) to postpone the condition evaluation or action execution so that they are not performed immediately, which explains why several different rule execution models exist. Thus the condition does not have to be evaluated or the action executed immediately after the event has been detected and the condition evaluated, respectively, but some time can pass in between. As a result, the condition can be evaluated at the end of the triggering transaction or the action can be executed in a new transaction, which does not depend on the triggering one. More on different execution models can be found in [6] and [17].

Very important question concerning ADBMS is the static analysis. Namely, active databases (occasionally) do not exhibit the desired behaviour; on the contrary, the behaviour could be described as undesired or even unpredicted. There is a possibility that triggers will trigger one another, that triggers are redundant or even inconsistent, and that the system behaves strange. Because of that it may happen that the rule processing will never terminate or the system will even crash. Further on, a single event can trigger several rules; which rules should be executed and in what order is also an important question. Therefore, several different approaches have been introduced in order to check redundant and/or inconsistent rules, and to determine whether the rule execution process terminates. We have addressed already mentioned issues in [6]; several approaches that can resolve these flaws were mentioned including meta-rules, knowledge based techniques, graph theory, etc.

Each active database management system is based upon a passive, conventional database management system. In order to support active functionality each passive database management system has to be extended in a way that different kinds of events can be detected, transactions can be managed because of different rule execution models, etc. A passive DBMS can be extended using integrated, layered or application oriented approach, as can be found in [17] and [19]. Due to this difference some tools for performance measurements have been introduced too ([22]).

There are several arguments justifying the use of ADBMSs. First of all, it is cheaper to build such application and its performance is better, at least when a small number of triggers is involved [17]. Secondly, such an application is smaller and easier to maintain, as is described in [5]. Thirdly, triggers tend to be considered as declarative technology and, according to [2] the trend has clearly always been away from procedural and toward declarative – that is, from how to what.

Triggers were introduced in 1970s, and since then they were used in many different fields for accomplishing many different tasks. During the years they became more complex and nowadays it is relatively hard to write triggers efficiently because each system has some peculiarities and is not in accordance with the SQL standard [6]; that is why some techniques were introduced in order to write triggers more easily. One interesting approach is so-called Trigger-By-Example approach that is based on Query-By-Example (QBE) idea ([3]). This approach makes it possible that triggers are built graphically using the main ideas of the QBE approach i. e. by filling the table structure with proper symbols.

Even today, active databases are used in many different areas, as can be found in [17]. We have already used active databases in order to test and see what are the benefits of their usage when implementing different kinds of business rules ([5], [7]). Further on, we have investigated the connection between business rules, active databases and reactive agents and published the results in an original scientific paper [9], and we have just presented the idea of how to build a multimodal biometric system using triggers in [8].

More on active databases can be found in [2], [5], [6], [17], and [23].

IV. Complex events and authorization

As it can be seen in already presented biometric models (Fig. 1, Fig. 3), sample processing and decision-making are performed out of the database. The module responsible for comparison and decision-making is placed out of the database and the system operates as a passive application [8]. It means that data must be extracted from the database and data processing (as well as decision-making) is performed out of the database. One has to have in mind that, when discussing multimodal biometric systems, even more data has to be extracted. So basically, while the database contains the data, an extra tier is added that is responsible for authorization. In such a case time needed for authorisation is significantly bigger because the data has to be pulled out from the database; one has to have in mind that we are not pulling out just integers or characters, but large amount of data (Binary Large OBjects) that makes the whole process very much dependent on network and database performances. In our solution this extra tier is being removed, and the multimodal biometric system functionality is being implemented and placed within the database.

One can conclude that existing biometric systems are passive applications, and that it makes sense to try to convert them into active applications, as we have already described (active applications possess better performances, triggers are easier to maintain, etc.).

So, the main idea is to place the decision-making module within the database and to express the functionality of multimodal biometric system (borrowing terms used in the active database theory) as a *real-time complex event detection*. We define a complex event that consists of several (*n*) simple events and each simple event represents a fact that the user was identified by means of one biometric feature that multimodal biometric system comprises of. Instead just to store the data in the database, the idea is to place the logic into the database as well. It has been already mentioned that speed is very important when talking about biometric systems, and we hoped to develop a better (faster) solution. The rest of the paper will show how this was done, and present some preliminary results.

Since a multimodal biometric system uses several biometric features in order to authorise a person (*n* features), we have defined a complex event *User_authorization* that consists of *n* simple events and could be written as follows:

$$User\_authorization = Biometric\_feature_1 \wedge Biometric\_feature_2 \wedge \ldots \wedge Biometric\_feature_n$$

So, the user is authorised if he has passed all biometric features checks (the first biometric feature, the second biometric feature, and so on, including the last biometric feature). For example (we have already said that it is not so easy to select which features are to be included in the system), more concrete,

User_authorization = *Thermogram* ∧ *Voice* ∧ *Facial_shape*

Occasionally, user has to be authorized within some time interval *t*. In that case we can construct a complex event *C* using the complex event constructor *within* which denotes that some event (simple or complex) must occur within time interval *t*. So, complex event *C* could be written as:

C = *User_authorization* within *t*

where the event *User_authorization* has been already defined.

The following picture represents the new model; comparison (authorization) is performed within the database.

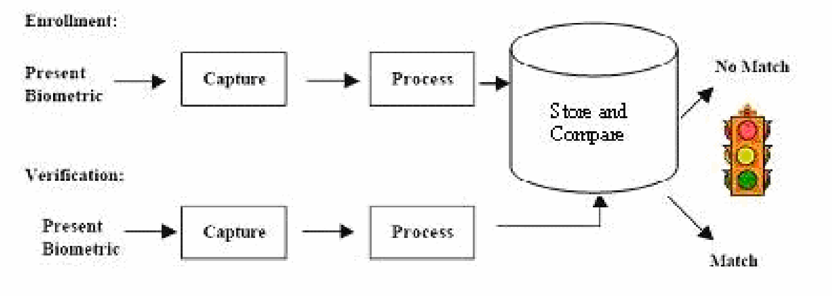

Figure 5. Enrolment and Verification – the new approach

We have built two solutions (application- and trigger-based) to test the proposed model. We have built a WEB application that stores biometric traits into the database and tries to authorise (identify) the user, and we have implemented triggers (and procedures) based upon complex events and constructor *within*, which behave as already described. The constructor *within* was not supported (we used PostgreSQL DBMS) so we had to implement it.

So here is what happens; when some feature (trait) is extracted, it is stored into the database. After the storage process, a query is posed to the database that tries to discover whether we can identify the user or not, or to check whether it is possible to find a user to whom the extracted data belong. When some biometric trait is stored into the database, triggers are triggered and they also try to authorise (identify) the user. The database contains small number of traits for now; this affects the results but does not represent a problem because the proposed solution is general and can be used within databases which contain much more data; after all, it was the idea. Time was measured for both solutions and represents how long did the decision-making process (identification) last. We have tested the proposed model using two and three biometric features. Based upon our experience with the active databases, we expected to reduce the time needed for the authorization.

We have presented Fig. 6 as well, because it clearly delineates the main modules of a biometric system. Once more one can see that decision-making module is placed out of the database.

Table I presents yielded results for ten randomly selected successful identifications in milliseconds for two biometric features, and Table II presents yielded results for ten randomly selected successful identifications in milliseconds for three biometric features.

TABLE I
WEB vs. Trigger-based solution (two biometric features)

Attempts	WEB-based solution (ms)	Trigger-based solution (ms)
1.	7,699	0,592
2.	7,721	0,591
3.	7,689	0,625
4.	7,761	0,809
5.	7,685	1,091
6.	7,759	1,117
7.	7,703	1,066
8.	8,548	0,603
9.	7,834	0,6
10.	7,755	0,609
On average:	7,815	0,77

In order to ensure that results are objective, no other scheduled tasks were running during the measurements. When the biometric trait was stored into the database, the application solution was stopped (for a few seconds) in order to avoid that the database is being queried while triggers were fired. When the trigger solution was finished, the application solution started to query the database in order to authorise the user.

In our solution we haven't used different extraction algorithms in order to analyse biometric features, extract the data and store the data into the database, but we have reduced the problem and used the results of extraction algorithms that, based on a biometric feature, produce sequences of bits that (in fact) represent the same feature. So, we didn't put accent on extraction algorithms, but we have used already extracted biometric data in order to test the functionality of the proposed model (we have abstracted the problem).

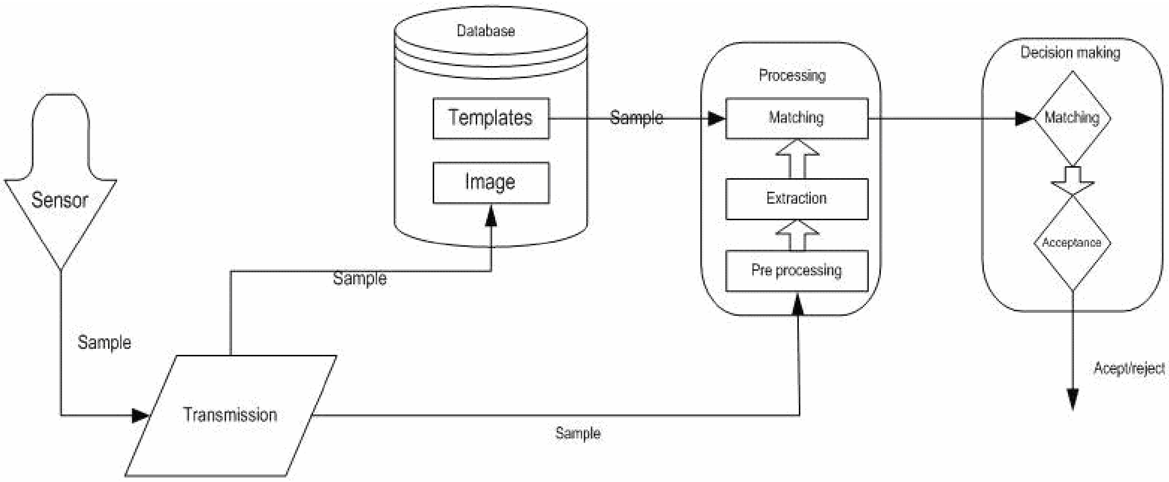

Figure 6. The main modules of a biometric system [10]

TABLE II
WEB vs. Trigger-based solution (three biometric features)

Attempts	WEB-based solution (ms)	Trigger-based solution (ms)
1.	7,995	1,661
2.	8,004	1,437
3.	7,997	1,561
4.	8,023	1,455
5.	8,453	1,559
6.	8,011	1,530
7.	8,665	1,499
8.	8,051	1,480
9.	8,047	1,486
10.	8,143	1,606
On average:	8,1389	1,5274

We have also tested the proposed model and compared the gained results for three biometric features as well. The results are presented in Table II.

Tables I and II present the obtained results. During the first attempt in Table I the trigger based solution required only 0,592 ms, while the application based solution required 7,699 ms, and so on (other rows can be read in the same way). The average time needed for the identification for the WEB solution based upon the data in the Table I was approx. 7,815 ms, while trigger based solution required only 0,77 ms in average. As we can see in the first table, trigger based solution requires about ten times less time (on average) in order to identify a person. It is usually said that trigger execution time is small and can be neglected (almost 0 sec.); that is why we were expecting to reduce the needed time. Identification is time-consuming process because the extracted feature has to be compared with all stored features (templates) in the database; so the idea of how to reduce the execution time is of a great interest.

In Table II presented rows can be interpreted in the same way, and the only difference is that three biometric features were included. As it can be seen in Table II (on average), for the trigger-based solution time has increased a little bit more than for the WEB-based one.

The trigger-based solution operates using the following algorithm:

```
After the trait is stored into the database
Try to find a user (id) whose trait was stored
If successful then
    Determine the current time (T2) and time – 10 minutes (T1)
    Try to find the user's other traits between T1 and T2
    If successful
            User is authorised
    If not
            User cannot be authorised
If not
    User is unknown
```

These are just experimental (preliminary) results, but for testing purposes in presented examples it was shown that triggers are useful and have provided desirable behaviour.

V. Conclusion

Active databases have shown again that the capability of automatic reaction to some events is of a great potential, and in this case it was demonstrated within the proposed model of multimodal biometric system. Time required for execution of certain actions is very small, network load is reduced and all constraints are collected and written in just one place. In this paper we have shown that active databases represent much faster solution for multimodal biometric system implementation then an application solution doing the same work. Preliminary results were presented and for now the idea seems very promising.

Our solution represents a novel model, and conducted preliminary experiments show that the time needed for authorisation has been reduced. Once more it has been shown that the concept of triggers is of great importance, and that some existing problems can be solved efficiently. So the application field of active databases has been extended once more.

Future research will include the probability model as well; for now the decisions were binary (just yes or no). Further on, because of their great importance, we will also consider FAR (False Acceptance Rate) and FRR (False Reject Rate).

REFERENCES

[1] Jain, R. Bolle and S. Pankanti, Biometrics: Personal Identification in Networked Society, Kluwer Academic Publishers, Norwell Massachusetts, 1999.

[2] J. Date, What not how – the business rules approach to application development, Addison Wesley, 2000.

[3] Lee, W. Mao, H. Chiu and W. W. Chu, Designing triggers with Trigger-By-Example, Knowledge and Information Systems, vol 7., 2004.

[4] F. L. Podio and J. S. Dunn, Biometric Authentication Technology: From the Movies to Your Desktop, Available from http://www.itl.nist.gov/div893/biometrics/Biometricsfromthemovies.pdf, 2001.

[5] K. Rabuzin and M. Maleković, Implementing business rules in active databases, The Proceedings of 15th International Conference on Information and Intelligent Systems, Faculty of Organization and Informatics, pp. 3-9, 2005.

[6] K. Rabuzin, M. Maleković and A. Lovrenčić, The active database theory and the SQL standard, The Proceedings of 18th International Conference on Information and Intelligent Systems, 2007, in press.

[7] K. Rabuzin, Aktivne baze podataka: implementacija poslovnih pravila u PostgreSQL-u, Master of science thesis, Faculty of organization and informatics, Varaždin, 2004. (in Croatian).

[8] K. Rabuzin, M. Maleković and M. Bača, A multimodal biometric system based upon the active database paradigm, Journal of Management, Informatics and Human Resources, vol. 39, no. 7, Kranj, Slovenia, pp 425-431, 2006.

[9] K. Rabuzin, M. Maleković and M. Bača, Active databases, business rules and reactive agents What is the connection?, Journal of Information and Organizatinal Sciences JIOS, vol. 29, no. 1, Varaždin, Croatia, pp. 63-73, 2005.

[10] M Bača and K. Rabuzin, Biometrics in Network Security, The Proceedings of the XXVIII International Convention MIPRO 2005, Rijeka, pp. 205-210 , 2005.

[11] M. Bača and K Rabuzin, A symbolic database for fingerprints storage - how to build it?, Proceedings of 8th Spring International Conference ISIM'05 Information Systems Implementation and Modelling, Ostrava, Czech Republic, pp. 295-302, 2005.

[12] M. Bača, K. Rabuzin and Z. Merkaš: Fingerprints preprocessing using walsh functions, Journal of Information and Organizational Sciences JIOS, vol. 30, no. 1, Varaždin, Croatia, pp. 1-12, 2006.

[13] M. Bača, M. Schatten and K. Rabuzin, Framework for sistematization and categorization of biometric methods, The Proceedings of 17th International Conference on Information and Intelligent Systems, Varaždin, Croatia, pp. 271-278, 2006.

[14] M. Bača, Ž. Hutinski and K Rabuzin, Palm Recognition and Security, Information Systems Security, MIPRO 2006, 29th International Convention, Rijeka, Croatian Society for Information and Communication Technology, pp. 172-175, 2006.

[15] M. Žugaj, J. Šehanović and M. Cingula, Organizacija, Tiva tiskara, Varaždin, 2004.

[16] N. Solayappan and S. Latifi: A Survey of Unimodal Biometric Methods, University of Nevada, USA, http://ww1.ucmss.com/books/LFS/CSREA2006/SAM3155.pdf

[17] N. W. Paton, Active rules in database systems, Springer, New York, 1998.

[18] R. Brunelli and D. Falavigna, Personal identification using multiple cues, Pattern Analysis and Machine Intelligence, vol. 17, no. 10: 955 – 966, 1995.

[19] S. Chakravarthy, Architectures and monitoring techniques for active databases: An evaluation, Data & Knowledge Engineering, vol. 16, no. 1, pp. 1-26, 1995.

[20] S. Prabhakar and A. K. Jain, Decision-level fusion in fingerprint verification, Pattern Recognition, vol. 35, no. 4, pp. 861 – 874, 2002.

[21] S. Prabhakar, S. Pankanti and A. K. Jain, Biometric Recognition: Security and Privacy Concerns, Biometrics, Mar/Apr 2003: pp. 33–42, 2003.

[22] U. Cetintemel, J. Zimmermann, Ö. Ulusoy and A. Buchmann, OBJECTIVE: a benchmark for object-oriented active database systems, Journal of Systems and Software, vol. 45, no 1, pp. 31-43, 1999.

[23] X. Li, J. M. Marin and S. V. Chapa, A Structural Model of ECA Rules in Active Database, Coello et al. (Eds.): MICAI 2002, LNAI 2313, Springer-Verlag Berlin, pp. 486–493, 2002.

Kornelije Rabuzin was born in 1979. in Varaždin, Croatia. He studied Information Technology at University of Zagreb, Faculty of Organization and Informatcs, Varaždin, where he graduated in 2001 and was promoted as the best student in the class. During his study, he was awarded twice as one of the best students. After the graduation, he continued working at Faculty of Organization and Information Science in Varaždin as an assistant. Currently he is holding exercises in Databases and Programming I. In 2003, he received an award as the best young assistant at the Faculty. He received a MSc. degree in 2004, and PhD degree in 2007. His fields of interests include (active) databases, biometrics and agent systems. He worked on several projects concerning e-learning, biometrics and multiagent systems.

Mr. Rabuzin has published more than 20 scientific and professional papers. He is a member of organizing committee of IIS conference, and a member of editorial board of JIOS (Journal of Information and Organizational Sciences).

Miroslav Bača was born in 1970. He received his MSc. degree in Information science in 1999. and PhD. degree in 2003, both issued by University of Zagreb, Faculty of Organization and Informatics, Varaždin. His fields of interests include biometrics, security systems and computer crime. He is currently an assistant professor at the University of Zagreb, Faculty of Organization and Informatics, Varaždin. He has finished several courses related to computer system management, database searching and e-learning.

Mr. Bača is currently a member of a program and organizing committee of International Conference POWA, organizing committee chairman of International Conference on Information and Intelligent systems (IIS). He published two books: Police and Security (in Croatian) and Introduction in Computer Security.

Mirko Maleković is a full time professor of Information Science, at University of Zagreb, Faculty of Organization and Informatics, Varaždin. Key responsibilities: Head of the Department of the Theoretical and Applied Foundations of Information Science. Professor of Data Bases (undergraduate); Relational Data Bases, Object Data Bases, and Deductive Data Bases (postgraduate). He serves as a member of Steering Committee of INES (International Conference on Intelligent Engineering Systems, IEEE).

Mr. Maleković is also a member of Program Committee of the following international conferences: ICEIS (International Conference on Enterprise Information Systems) and IIS (International Conference on Intelligent Information Systems). He serves as an editor of JIOS (Journal of Information and Organizational Sciences), and a member Editorial Board of ComSIS (Computer Science and Information Systems). Research interests include databases, knowledge bases, semantic modelling, reasoning about knowledge and multi-agent systems.