CIHSPS2004 - IEEE International Conference on
Computational Intelligence for Homeland Security and Personal Safety
Venice, Italy, 21-22 July 2004

A High-level Optimum Design Methodology for Multimodal Biometric Systems

Marco Gamassi, Vincenzo Piuri, Daniele Sana, Fabio Scotti
Department of Information Technologies, University of Milan
Via Bramante 65, 26013 Crema (CR), Italy
Phone: +39-02-503-30066, Fax: +39-02-503-30010, Email: {gamassi, piuri, sana, scotti}@dti.unimi.it

Abstract *– Biometric systems are designed by expert developers who look – with trial-and-error approaches – for reasonable solutions by considering the available hardware architecture, some –possibly conflicting– quality goals, and the application constrains. Typically drawbacks of these approaches are waste of time and results far from the optimum.*

In this paper we propose a new design methodology for multimodal biometric systems that applies high-level system design techniques to better structure the design procedure. The proposed methodology avoids the drawbacks of the common design practice and allows to create a flexible general-purpose and effective design environment for multimodal biometric systems.

Keywords *– multimodal biometric systems, design of biometric systems, system evaluation, high-level design.*

I. INTRODUCTION

Biometric systems are defined as systems exploiting "automated methods of recognizing a person based on physiological or behavioural characteristics" (*biometric identifiers*, also called *features*) [1]. Physiological biometrics is based on data derived from direct measurement of a body part (e.g., fingerprints, face, retina, iris), while behavioural biometrics is based on measurements and data derived from a human action (e.g., gait and signature) [2].

Biometric systems are composed by one or more sensors included into an embedded system, or connected to a PC , or to a distributed system. Examples of embedded biometric systems are biometric door locks and cellular phones with biometric authentication system for credit card transactions. Examples of distributed biometric systems are biometric authentication systems in airports with many sensors units and a shared biometric traits database placed in another location.

If a biometric system exploits one single biometric trait is called *monomodal*, otherwise it is *multimodal* (Figure 1) [1]. The computation performed by a monomodal biometric system is divided into three cascaded actions:

- *filtering*, to enhance the quality and the readability of the biometric samples;
- *feature extraction*, to extract relevant features from the biometric samples;
- *matching*, to compare the features extracted from the samples to the ones of persons stored in a database.

The cascade Filtering / Feature Extraction / Matching is called *biometric chain*. The matching produces a matching value that is evaluated by the decision algorithm in order to produce the *authentication answer* (authorized/not-authorized). These components are described at high-abstraction level by algorithms.

Multimodal systems acquire different biometric traits and have a complete chain for each type of acquired trait. For example, a tri-modal system can have in input fingerprint, voice, and face samples. Each chain produces a matching value for each biometric trait. These values are collected by the decision algorithm, which then produces the authentication answer.

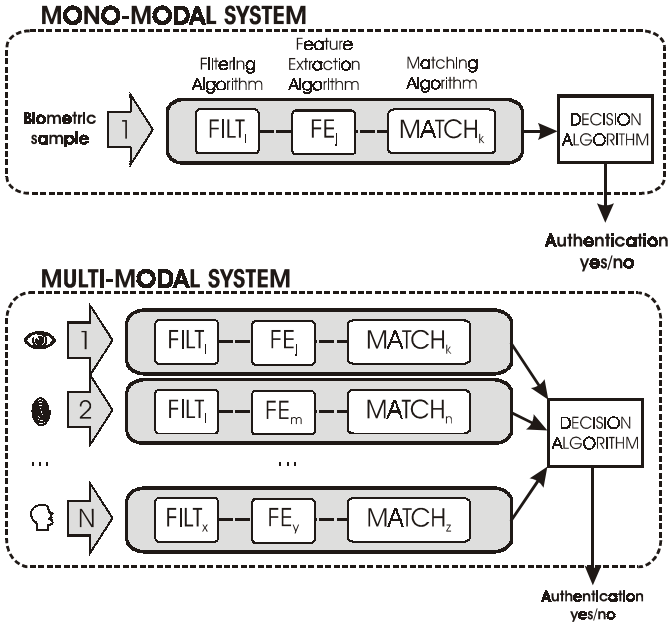

Fig. 1. Processing structure for monomodal and multimodal biometric systems

Nowadays, the algorithms for a biometric system are designed by expert developers, who try to satisfy given requirements by taking into account the available hardware architecture. The design process looks for a trade-off among requirements (e.g., accuracy, computational complexity, memory usage, system cost), that are competitive against each others. On the other hand, biometric hardware architectures are very often required to be small and low-power consuming (e.g., cellular phones and smart cards with biometric authentication).

The trade-off among requirements is typically achieved by a trial-and-error, which leads to wasting time and results in solutions often far from the optimum (Figure 2).

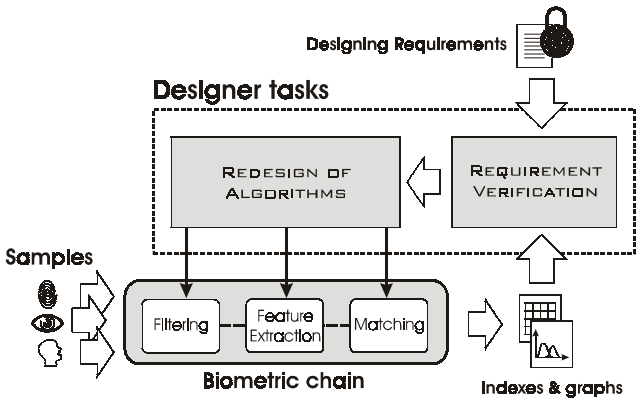

Fig. 2. Iterative trial-an-error approach for biometric system design

Differently from the classical approach, in this paper we propose to apply the knowledge available in high-level system design [21, 22] to the semi-automatic design of biometric systems. This paper deals in fact with the choice of algorithms to be inserted into the biometric system and the optimization of the hardware system architecture implementing these algorithms. The output produced by our methodology is a ready-to-compile code and a suitable configuration of the hardware architecture.

The paper is structured as follows. In section II we describe the methodology, its inputs, and the generated output, and how to apply the methodology into application cases. Then, the system optimization is described. Section III describes a prototype implementation of the methodology by means of an object-oriented toolbox and the experimental results.

II. A NEW DESIGN METHODOLOGY

Current designing procedures for biometric chains have evident drawbacks (e.g., time consuming, non optimality). These negative aspects can be avoided by a more comprehensive approach that aims to automate the trivial and repetitive design tasks and to allow designers to better explore the design space. The proposed methodology can produce benefits such as results closer to the optimum than traditional trial-and-error approaches and time saving in the design process. In addition, the design experience of experts can be shared easier and in an automated way. These goals can be effectively achieved by means of high-level design techniques.

High-level synthesis is the process of mapping a behavioural description at the algorithmic level to a structural description in terms of functional units, memory elements, and interconnections [41]. The term *behavioural description* refers to a description of the input/output relationship of the system to be implemented. This is typically given by means of an algorithm written, e.g., in C, C++ , VHDL, and System C.

The proposed methodology can be considered as a *high-level optimum synthesis* approach and can be summarized in the three following main activities:

(1) to *model* the possible *hardware architectures* that are available in the design environment to implement the biometric systems;
(2) to *specify* the *behavioural description* of the biometric system for the envisioned application, by including both the functional description and the non-functional characteristics and constrains;
(3) to *map* the behavioural description for the specific application into a hardware model with the given non-functional constrains by means of an iterative procedure composed by the following operations:
   a) the behavioural description is mapped onto the hardware architecture by satisfying the designer's requirements;
   b) proper figures of merit are evaluated by considering the current system;
   c) system's independent variables are tuned in order to better satisfy the designer requirements, by searching for an optimal solution achieved by iterating these three operations (multi-objective constrained optimization).

In our case both the hardware description and the behavioural description can contain independent variables to be optimized during the mapping phases. For example, the biometric algorithm given as behavioural description can be parameterized: its parameters will be added to the independent variables of the biometric system to be mapped.

Figure 3 shows the overall methodology. Once the behavioural description, *A*, of the biometric system is selected, it is mapped onto the hardware architecture model, *HW*, thus producing the complete model of the complete biometric system, *bio=HW(A)*. Then, proper figures of merit (*figures*) are computed by using the obtained *bio* system.

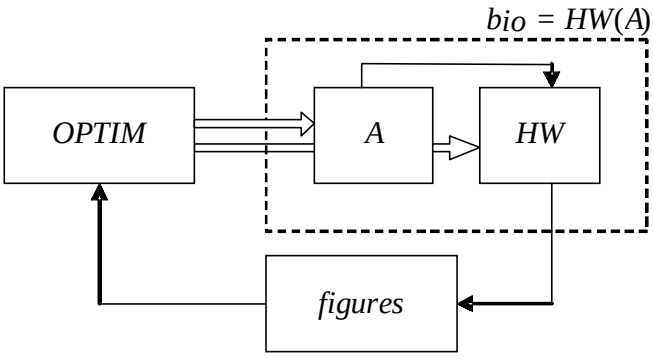

Fig. 3. Application of the methodology. *A*: behavioural description; *HW*: hardware architecture model; *figures*: processed figures of merit based on the biometric system *bio*; *OPTIM*: optimization system.

The optimization engine *OPTIM* changes the independent variables of the behavioural description and the independent variables of the hardware architecture model in order to

enhance the figures of merit by following the policy given by the designer. Let us now detail all proposed steps and components.

A. The hardware architecture model

The hardware architecture model, *HW*, aims to describe the available hardware supporting the biometric system and its characteristics. The hardware architecture model we propose describes a *general* – possibly distributed- biometric system, composed by a collection of suitably interconnected hardware modules (e.g., dedicated or configurable integrated circuits, dedicated boards, personal computers, servers). The model can contain design parameters such as the topological graph of the hardware architecture modules, their interconnections, cost and computational performances.

The basic idea is to describe where and how each single module of a multimodal biometric system can/must be implemented. The model can be easily upgraded and it is flexible. For example, by setting the graph parameters, the model can effectively describe an *embedded system*: in this case all modules belong to the *same* hardware board. At the other extreme, also *distributed systems* can be described: as an example, let us consider a multimodal system in an airport where a multimodal sensor unit queries a remote database of biometric data.

The general schema of the hardware architectural model is shown in Figure 4. *N* sensors are used to acquire *N* traits of an individual (biometric *samples*). Samples are processed by the filtering algorithms ($FILT_1$, … $FILT_N$) and then the feature extraction algorithms (FE_1, … FE_N) produce sets of extracted *features*. These sets are used as input for the *N* matching algorithms. These algorithms can query different databases containing reference features (called biometric *templates*), possibly also located in different sites. Each matching algorithms computes a matching score. All the matching scores are transferred to a *decision algorithm*, which produces the authentication value.

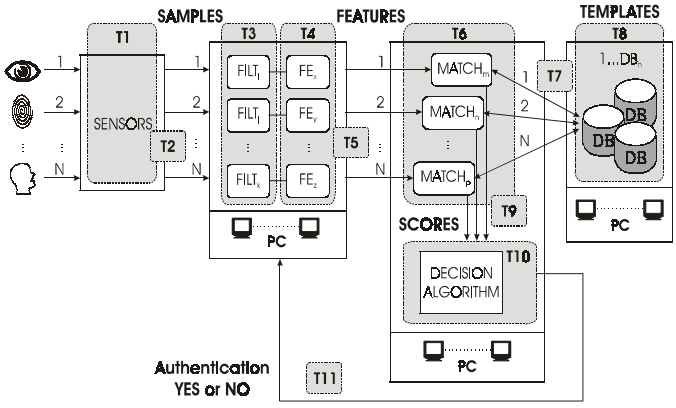

Fig. 4. Hardware architecture model
of a multimodal distributed biometric system

Filtering and feature extraction may process on different hardware boards, thus considering algorithms pipelining to increase the system throughout. The *N* chains are independent and, hence may be parallel architecture to enhance the system performance. A multimodal biometric system can be built by grouping together commercial-of-the-shelf components for monomodal systems. Our model describes also these configurations.

Each algorithm described in our model, can be implemented into a specific hardware module (e.g., PC, board, integrated circuits), labelled by a unique numeric identifier *k*; the type of the hardware module is identified by PC_k.

The allocation of each algorithm which is present in the model is described by a the mapping function *P(AL)* whose value is the identifier *k* of the hardware module on which the generic algorithm *AL* is be implemented. A possible implementation of the mapping function consists of a mapping table, as described in section II.C.

Since each algorithm requires a specific amount of memory to be processed (if not directly implemented in hardware), the total memory available for applications to be installed in the *k*-th hardware module must be equal to the sum of the memory hosted on this unit if all algorithms must reside in the memory at the same time. Less memory may be sufficient if algorithms run serially by adopting a memory overlaying approach, or are executed in a time-sharing environment by using a virtual memory management approach. Performance will change according to the adopted memory management strategy.

Execution time of each algorithm is also considered: **T1**, **T3**, **T4**, **T6**, **T8** are vectors containing the execution times for sample acquisition, filtering, feature extraction, matching, DB querying, of each biometric chain, respectively. **T10** is the time that the decision algorithm needs to process the authentication decision. Since the system can be distributed, data transfer time is considered. **T2**, **T5**, **T9** are vectors of time needed to transfer samples, features, and matching score between parts, respectively. **T7** is the vector of maximum time requested to query the databases. Finally, **T11** is the transferring time of the authorization value to the input units.

In our methodology the environment includes all hardware architecture models that may implement any possible application. All described configurations of the hardware architectures have to be collected into a *hardware architecture database*, DB_{HW}. During the optimal mapping, the optimization engine will test possible hardware configurations by exploring this database and by taking into account the designer requirements.

B. The behavioural description

The behavioural description, *A*, of the biometric system consists of the *sequence of the operations* that allow the biometric system to identify the person presented at its input sensors. The behavioural description is typically stated by the designer by means of a list of *algorithms* and by specifying how they must be combined into the computational flow of the

biometric system. More in detail, our behavioural description represents both monomodal and multimodal biometric systems in terms of the processing architecture (Figure 1) and which algorithms will be used in the structure.

We assume that the designer collects the available algorithms in the *algorithm databases*, namely the filtering algorithms $DB_{Filtering}$, the feature extraction algorithms $DB_{FeatureExtraction}$, and the matching, and $DB_{Matching}$.

During the optimization of the biometric system, the optimization engine will produce different compositions of algorithms by exploring these databases. Different releases of the same algorithm may be present in the databases. We assume that all algorithms are parameterized: each algorithm has a number of parameters that can affect its operation, accuracy, and computational complexity: default values for each algorithm are provided.

A behaviour description of a multimodal biometric system can thus be compactly given as follows:

$$S = S(C, DC) = S\left(\begin{bmatrix} AL_{11}(\bar{\theta}_{11}) & AL_{12}(\bar{\theta}_{12}) & AL_{13}(\bar{\theta}_{13}) \\ AL_{21}(\bar{\theta}_{21}) & AL_{22}(\bar{\theta}_{22}) & AL_{23}(\bar{\theta}_{23}) \\ \dots & \dots & \dots \\ AL_{N1}(\bar{\theta}_{k1}) & AL_{N2}(\bar{\theta}_{N2}) & AL_{N3}(\bar{\theta}_{N3}) \end{bmatrix}, DC(\bar{\theta}_{DC})\right) \quad (1)$$

$$\bar{\theta}_{ij} = \left[\theta_{ij}^{(1)}, \theta_{ij}^{(2)}, \dots, \theta_{ij}^{(n_{ij})}\right]; \quad \bar{\theta}_D = \left[\theta_D^{(1)}, \theta_D^{(2)}, \dots, \theta_D^{(n_D)}\right] \quad (2)$$

$$\begin{aligned} AL_{i1} &\in \{F_{i1}, F_{i2}, \dots, F_{iX_i}\} \subseteq DB_{Filtering} \\ AL_{i2} &\in \{FE_{i1}, FE_{i2}, \dots, FE_{iY_i}\} \subseteq DB_{FeatureExtraction} \\ AL_{i3} &\in \{Match_{i1}, Match_{i2}, \dots, Match_{iW_i}\} \subseteq DB_{Matching} \\ DC_d &\in \{Decision_1, Decision_2, \dots, Decision_Z\} = DB_{Decision} \end{aligned} \quad (3)$$

where C is Nx3 matrix of biometric chains algorithms, AL_{i1} is a a filtering algorithm, AL_{i2} is a feature extraction algorithm, AL_{i3} is a matching algorithm.

For example, the filtering algorithm F_{ij} corresponds to the j-th filtering algorithm available for the i-th biometric modality. Similarly we use the same notation for the feature extraction algorithm FE_{ij} and the matching algorithm $Match_{ij}$.

Each algorithm $AL(.)$ is a function of the configuration parameters' vector $\bar{\theta}$; in equation (2) $\bar{\theta}_{ij}$ is the vector of parameters that characterizes the j-th algorithm available for the i-th biometric modality.

Equation (1) indicates with DC the decision algorithm and with $\bar{\theta}_D$ its vector of parameters. The decision algorithm combines the matching scores produced by the different biometric modalities to generate the authorization output. Also the decision algorithm DC is a function of the configuration parameter vector $\bar{\theta}_d$. Decision algorithms are stored in the $DB_{Decision}$ database.

Common strategies implemented in the decision algorithm include majority voting, product rules, k-NN classifiers, SVMs, decision trees, and Bayesian methods [35-40]. In monomodal systems the decision algorithm simply implements a threshold for the scores provided by the single matching algorithm. In multimodal biometric systems the simplest form of combination would be to take the thresholded weighted average of the scores from the multiple modalities.

C. *Mapping the behavioural description onto the hardware model*

The goal of the *mapping phase* consists of binding each component of the behavioural description A to the corresponding hardware resources HW, which implement its computation in the biometric system. Mapping must take into account the requirements given by the designer.

The optimum mapping is an iterative process in which proper figures of merit are evaluated by considering the current system, and in which system's independent variables are tuned to enhance the system's figures of merit while satisfying the design requirements. The multi-objective optimization process is repeated by searching for optimal solution. The figures of merit and the optimization issue are detailed in the next sections.

It is worth noting that the mapping procedure is not trivial since not all of the algorithms, which are in databases can be combined into the biometric system. Hence, the composition of algorithms are present into the behavioural description should not be randomly explored since random combinations of algorithms may not be compatible solutions according to constrains. Precise rules must be followed by taking into account the semantic of each component. In addition, algorithms of different chains (for example two matching algorithms) cannot be interchanged due to possible different syntactic characteristics of the interface (e.g., input/output data-types).

The relationship between the behavioural description and the hardware model can be implemented, for example, by using *mapping tables*, while the semantic and input/output analysis can be implemented by means of a *rule-based system*.

Table I shows an example for a single-board embedded monomodal biometric system. The table permits to locate each single algorithm belonging to the behavioural description A of the biometric system into the hardware model HW. The first column identifies the algorithm; the second column contains the algorithm's name contained into the algorithm database. The third column locates the hardware module present in the hardware architecture, which executes the considered algorithm.

TABLE I
Mapping for a single-board embedded monomodal biometric system

Algorithm ID	*Algorithm NAME*	*LOCATION*
AL_{11}	$FILT_1$	PC_1
AL_{12}	FE_{12}	PC_1
AL_{13}	$MATCH_{12}$	PC_1
D	KNN_1	PC_1

More complex architectures can also be represented by using the mapping table. For example, Table II shows the mapping table for a distributed *bi*modal system that uses 3 separated PC (one for each biometric chain and one implementing the decision algorithm).

TABLE II
Mapping for a distributed multimodal biometric system

SW MODULE ID	*Algorithm NAME*	*LOCATION*
AL_{11}	$FILT_1$	PC_1
AL_{12}	FE_{12}	PC_1
AL_{13}	$MATCH_{12}$	PC_1
AL_{21}	$FILT_{24}$	PC_2
AL_{22}	FE_{32}	PC_2
AL_{23}	$MATCH_{23}$	PC_2
D	KNN_2	PC_3

The knowledge about *how to combine* algorithms in biometric chains has been formalized by means of *rules*. The list of rules defines the semantic and syntactic compatibilities of the algorithms to be used to solve the application.

To support the selection of compatible algorithms a rule-based system can be adopted. The main advantages of the *rule system* are given by its flexibility and scalability; new algorithms can be easily inserted since the matrix of the achievable chains can be automatically reprocessed.

The goal of the rule-based system is to produce a list of admitted chains by correctly composing their algorithms by considering their semantic and synthetic characteristics.

Figure 5 shows a graphic representation of the *rule-based system*, which is composed by the following three *correlated* lists:

- the ***filtering list*** contains the available filtering algorithms and the corresponding class; two filtering algorithms belong to the same class if their outputs are semantically equivalent (e.g., Gabor filtering and band-pass filtering belong to class 1);
- the ***feature extraction list*** contains, for *each* feature extraction algorithm, the compatible *filtering* algorithms classes, the available *feature extraction* algorithms, and the corresponding class; two different feature extraction algorithms belong to the same class if their *outputs* are semantically equivalent;
- the ***matching list*** contains the compatible feature extraction algorithms classes for *each* available matching algorithm.

The final complete list of valid chains can be easily created by combining the items in the algorithm databases under the constraints given by the rules. Figure 6 shows the correct connections between classes of components (with dotted lines) in an applicative case. Each possible path that starts from the first list and ends to the last list represents a correct chain. Each chain can be stored by using the notation [F_i , FE_j , $MATCH_k$]. Hence, the output of the rule-based system is the N Mx3 matrix where M is the number of valid chains that has been found for N biometric traits that are included into the multimodal biometric system.

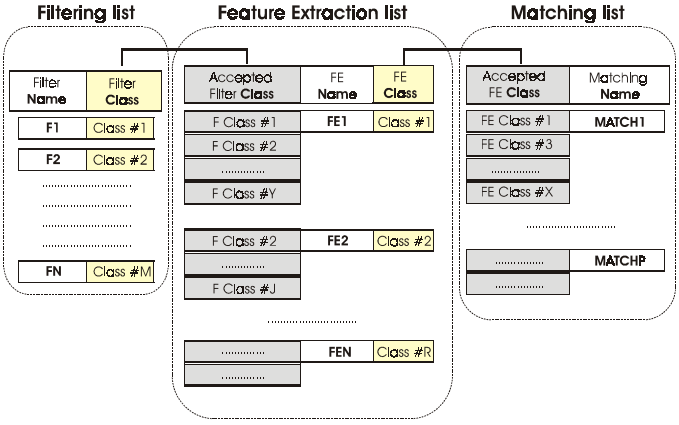

Fig 5. Biometric rule-based system

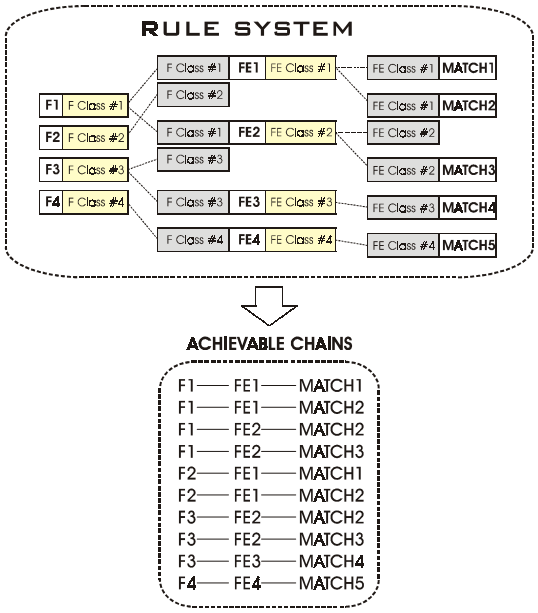

Fig. 6. Extraction of the valid chains from the rule-based system

D. Figures of merit for a multimodal biometric system

The most common figures of merit considered for a biometric system characterized its *accuracy*. Accuracy is usually evaluated by a set of indexes based on the concept of *error of misclassification*. Typically, the accuracy evaluation of the biometric system is performed by means of the procedure called *evaluation scenario* [13]. This procedure evaluates the accuracy indexes by considering a standard database of biometric features (e.g., a database of fingerprint images).

In the literature many indexes are used to reflect the effectiveness of a biometric system [8, 10-15, 20]. The *False Match Rate* (*FMR*) is the expected probability that a sample is falsely declared to match a single randomly-selected template (*false positive*). The *False Non-Match Rate* (*FNMR*) is the expected probability that a sample is falsely declared not to match a template of the same measure from the same user (*false negative*) [8]. Both indexes are functions of the threshold value *t* used to compare the matching value to make the decision.

Other indexes can complete the accuracy description. The *EER* (*Equal Error Rate*) is often considered. It is computed as the point where *FMR(t) = FNMR(t)*. *EER* must be often interpolated by the quantized data [14,15]. *ZeroFMR* is the lowest *FNMR* for *FMR = 0%* (ideal is *ZeroFMR = 0%*); *ZeroFNMR* is the lowest *FMR* for *FNMR = 0%* (ideal is *ZeroFNMR = 100%*);

The evaluation of the overall accuracy level of a biometric chain is often computed by considering two error plots. The first one is the Receiving Operating Curve (*ROC*), where $(1-FNMR)$ is plotted as a function of *FMR* for all available values of the threshold *t*. The second, and most used, one is the plot of *FNMR vs. FMR* in a logarithmic chart, called the *Detection Error Trade-off* (*DET*) plot. In order to select the best system, a comparison among the *DET* curves is to be done. The best system is the one with the *DET* curve below all the others. An overall explanation of the accuracy indexes commonly used in literature is provided in [8, 13].

Other figures of merit are related to the hardware architecture and the implemented algorithms, for example the response time and the memory usage [15].

The *response time* is the total time required to complete the biometric authentication. It includes the execution time and the transfer time of all the considered modules. We should also include the possible waiting time on time-shared processors. If the complete biometric system is implemented on a single board, transition times **T1**, **T5**, **T9** and **T11** can be typically ignored. In a distributed biometric system, transfer time must be considered.

Once the final hardware architecture of the biometric system is defined, each available algorithm belonging to the behavioural description *A* can be tested and its execution time can thus be evaluated. Of course *simulators* [23] of the final hardware architecture can be used. Since more than one module can be hosted on a single processing hardware module, the *total* computation load must be considered.

The *memory usage* is another important figure of merit for biometric systems. Given the model of the hardware architecture, proper profiling tools [23, 24] permit the evaluation of the memory usage of an algorithm. Again when more than one module is implemented on a single processing hardware module, the *overall* memory consumption must be considered.

Interestingly, some figures of merits are directly related *only* to the algorithms used (such as *accuracy*). These figures of merit can be calculated using a very simplified hardware architecture model. Just few information such as number of digits and the presence of floating point unit are required to evaluate the figures of merit. Others figures of merit depend *only* on the hardware architecture (e.g., transfer time between modules like **T2**, **T5**, **T7** and **T11**).

Of course, designers can add others figures of merits into the methodology (such as the economical cost of the system), until the description of biometric system *bio* is enough complete. This scalability is allowed by the structured organization of the methodology.

E. Design requirements

Given the biometric model $bio = HW(A)$ obtained by mapping the behavioural description *A* on the hardware architecture model *HW* and the data *benchData* required to test the system, it is possible to evaluate the figures of merit. We use the following notation:

$$[f_1, f_2, \ldots, f_m] = figures\left(HW(A), benchData\right) \tag{4}$$

where the function *figures* returns a vector of numerical features $[f_1, f_2, \ldots, f_m]$ which quantitatively describes the aspects of the biometric system that the designer wants to consider.

At this stage, the design requirements are expressed by the designer as a *set of equations in the figures of merit*:

$$h(f_1, f_2, \ldots, f_m) \le P \tag{5}$$

where *P* is the requirements vector.

Let assume for example that a designer considered the following figures of merit: *EER*, *zeroFMR*, *zeroFNMR*, *responseTime*, *memoryOccupation*. Hence, the design requirements can be a set of equations similar to the following:

$$\begin{cases} EER < 0.01 \\ zeroFMR < 0.02 \ AND \ zeroFNMR > 0.98 \\ responseTime < 2s \\ memoryOccupation < 4MB \end{cases} \tag{6}$$

In the given example we consider three figures of merit (*EER*, *zeroFMR* and *zeroFNMR*) related to the accuracy and two both related to the activities and the hardware architecture (*responseTime* and *memoryOccupation*). Last two values have to be processed by proper simulation and profiling tools.

The designer can also constrain the exploration of the available algorithms present into the $DB_{Filtering}$, $DB_{FeatureExtraction}$ and $DB_{Matching}$ databases to a subset of them. Similarly, the designer can reduce the exploration of the available hardware architecture contained into the DB_{HW} database to a reduced subset.

F. Optimization

For his/her specific application, the designer is assumed to specify, the behavioural description *A*, the benchmark data

benchData, the dataset of algorithms ($DB_{Filtering}$, $DB_{FeatureExtraction}$, $DB_{Matching}$ and $DB_{Decision}$), the database of hardware architectures DB_{HW} , the set of figures of merits [f_1, f_2, …, f_m], the designer requirements stated as a set of equation in [f_1, f_2, …, f_m], and an optimization function J (given as a function of the figures of merit).

The optimization process selects by means of an iterative approach the values of the design independent variables that provide the best value of the multi-objective optimization function.

Let us now describe the independent variables $\boldsymbol{X}$ of the optimization. The *first set* $\boldsymbol{X_1}$ of independent variables comes from the behavioural description. They characterize the algorithms that are chosen and mapped onto the hardware architecture *HW*. By using the proposed formalism, we have *N*x3 variables AL_{ij} in equation (1), which can assume the values specified in equation (2) (the D_1 domain). This set of variables is always present.

The *second set* $\boldsymbol{X_2}$ of independent variables comes from the specification of the decision algorithm. If the decision algorithm is fixed, hence no variable have to be added into the optimization model. If the decision algorithm is parameterized, all its free parameters and their ranges (the D_2 domain) become independent variables for the optimization engine. For example, we can consider as decision algorithm a kNN classifier where the *k* parameters are not fixed (e.g., $k \in [1,3,..15]$).

The *third set* $\boldsymbol{X_3}$ of independent variables have to be considered in the case in which the algorithms AL_{ij} stored into the databases are parameterized ($\bar{\theta}_{ij}$). In this case all parameters contained into the $\bar{\theta}_{ij}$ vectors have to be considered as design variables of the optimization problem with their ranges (the D_3 domain).

The *fourth set* $\boldsymbol{X_4}$ of independent variables have to be considered when different hardware architectural choices have to be described. We in fact to set which processing hardware module PC_k will execute the algorithm AL_{ij}. Furthermore we have to describe if the hardware architecture can exploit parallel processing. Let name the domain of these variables D_4.

A formalization of the optimization problem can therefore be stated as follows:

- the independent variables vectors and their ranges:

  $X_1 \subseteq D_1; \quad X_2 \subseteq D_2; \quad X_3 \subseteq D_3; \quad X_4 \subseteq D_4;$

- the figures of merit:

  $[f_1, f_2, \ldots, f_m] = figures\left(HW\left(A(X_1, X_2, X_3), X_4\right), benchData\right);$

- the constrains:

  $h_v(f_1, f_2, \ldots, f_m) \leq 0;$

- multi-objective optimization function:

  $J = g([f_1, f_2, \ldots, f_m])$.

Typically, in the literature, biometric designers consider simpler problems or make problems simpler by a priori settings the values of some design independent variables, but without any concern to optimization (e.g., they may limit the hardware solution or adopt algorithms in which some/all parameters are set).

III. EXPERIMENTAL RESULTS

To verify the feasibility and the usability of the proposed methodology, we implemented a prototype of the methodology by means of an object-oriented toolbox written in Matlab. The general descriptor of the biometric data as been introduced by means of a object oriented approach: this object is called *biodata*. The operations that compose the methodology have been implemented as *biodata*'s methods by using the polymorphic property. As such, the rule-based system is independent from the biometric data. Flexibility and scalability are thus guaranteed.

Since we aimed only to show the viability of the methodology, we did not cared for completeness, accuracy in evaluation, and exhaustiveness at present time. In this prototype we introduced well-known biometric chains for pre-filtering, feature extraction, and matching algorithm which are available in the literature. For fingerprint-based systems we included algorithm described in [25-30], while for iris-based system we adopted algorithm described in [31-34]. The rule-based system allowed to create of candidate chains. In the current prototype the implemented figures of merit are *EER*, *zeroFMR*, and *zeroFNMR*.

The optimization system has been implemented by using the Matlab environment: the algorithms stored in the databases have fixed parameters.

An automated performance reporting method has been implemented. The polymorphic method can plot, for example, accuracy graphs, *ROC* curve, tables of selected figures of merits, sample plotting, and comparison.

IV. CONCLUSIONS

The proposed methodology represents an effective approach to describe the design activities for biometric chains and to support optimization.

We showed that many repetitive and time-consuming design tasks can be automated by using high-level design approaches. The proposed methodology was implemented and tested in an object-oriented prototype toolbox by allowing to create a flexible and easy-to up-date design environment.

Future research will address the more detailed analysis and the evaluation of the hardware alternatives, extensions of the biometric algorithm, multi-objective optimization algorithms, more integrated multimodal operations, and multi-agent collaborative biometric environment.

REFERENCES

[1] http://www.nist.gov/srd/biomet.htm

[2] The Biometric Consortium (2002) 'Introduction to Biometrics', http://www.biometrics.org/html/introduction.html

[3] Gerhard Weiss, "Multiagent Systems, A Modern Approach to Distributed Artificial Intelligence", The MIT Press, 1999.

[4] M. N. Huhns, U. Mukhopadhyay, L. M. Stephens, and R. D. Bonnell. "DAI for Document Retrieval: The MINDS Project". In M. N. Huhns, editor, *Distributed Artificial Intelligence*. Pittman, London, 1987.

[5] N. R. Jennings, "Coordination Techniques for distributed Artificial Intelligence". In *Foundations of Distributed Artificial Intelligence*, pages 187-210. John Wiley & Sons, Inc., New York, 1996.

[6] N. R. Jennings, "Commitments and Conventions: The Foundation of Coordination in Multi-Agent Systems ". *The Knowledge Engineering Review*, 2(3):223-250, 1993

[7] A. Haddadi, "Towards a Pragmatic Theory of Interactions", Proc. International Conference on MultiAgent Systems (ICMAS), San Francisco, 1995.

[8] M. Gamassi, M. Lazzaroni, M. Misino, V. Piuri, D. Sana, F. Scotti, "Accuracy and Performance of Biometric Systems", Instrumentation and Measurement Technology Conference (IMTC), 2004.

[9] J.D.M. Ashbourn, 'Biometrics : Advanced Identify Verification: The Complete Guide', Springer-Verlag, 2000

[10] The Biometric Evaluation Methodology Working Group, 'Common Methodology for Information Technology Security Evaluation', 2002

[11] P. J. Phillips, A. Martin, W.M. Przybocki, "An introduction to evaluating biometric systems", IEEE computer , vol. 33, no. 2, February 2000

[12] V.S. Valencia, 'Biometric Testing: It's Not as Easy as You Think', Biometric Consortium Conference, Sept. 2003, Arlington VA USA.

[13] A. J. Mansfield, J. L. Wayman, "Best practices in testing and reporting performance of biometric devices", Version 2.01, Aug. 2002.

[14] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman and A.K. Jain, "FVC2002: Second Fingerprint Verification Competition", in proceedings 16th International Conference on Pattern Recognition (ICPR2002), Québec City, vol.3, pp.811-814, August 2002.

[15] FVC2004 Fingerprint Verification Competition: http://bias.csr.unibo.it/fvc2004/

[16] FVC2000 database: http://bias.csr.unibo.it/fvc2000/databases.asp

[17] CASIA database: http://www.sinobiometrics.com/casiairis.htm

[18] R. Michael McCabe, 'Standards for Certifying Biometric Accuracy', BIOCONS2003.

[19] G. R. Doddington, et.al. "The NIST speaker recognition evaluation: Overview methodology, systems, results, perspective", Speech Communication, 2000, 31(2-3), 225-254.

[20] K. V. Diegert, "Estimating performance characteristics of biometric identifiers", Proceedings of Biometrics Consortium Conference, San Jose, CA, June 1996.

[21] G.De Micheli and M.G.Sami, "Hardware/Software Codesign", Kluwer Academic Publishers, 1995.

[22] D.Gajski, N.Dutt, A.Wu, and S.Lin, "High-level Synthesis: introduction to Chip and System Design", Kluwer Academic Publishers, 1992.

[23] T. Mowry, M. S. Lam, and A. Gupta. "Design and evaluation of a compiler algorithm for prefetching". In Proceedings of the Fifth International Conference on Architectural Support for Programming Languages and Operating Systems, pages 62--73, Boston, MA, Oct. 1992.

[24] A. R. Lebeck and D. A. Wood, "Cache Profiling and the SPEC Benchmarks: A Case Study," IEEE Computer, vol. 27, pp. 15--26, October 1994. http://citeseer.ist.psu.edu/lebeck94cache.html

[25] http://www.itl.nist.gov/iad/894.03/databases/defs/nist_nfis.html, NIST Fingerprint ImageSoftware(NFIS).

[26] Tico, M.; Kuosmanen, P., "An algorithm for fingerprint image postprocessing", Signals, Systems and Computers, 2000. Conference Record of the Thirty-Fourth Asilomar Conference on , Volume: 2 , 29 Oct.-1 Nov. 2000 Pages:1735 - 1739 vol.2.

[27] Ying Hao; Tieniu Tan; Yunhong Wang; **"**An effecitve algorithm for fingerprint matching" TENCON '02. Proceedings. 2002 IEEE Region 10 Conference on Computers, Communications, Control and Power Engineering , Volume: 1 , 28-31 Oct. 2002 Page(s): 519 –522.

[28] Mital, D.P.; Eam Khwang Teoh; "An automated matching technique for fingerprint identification" Emerging Technologies and Factory Automation, 1996. EFTA '96. Proceedings., 1996 IEEE Conference on , Volume: 1 , 18-21 Nov. 1996 Page(s): 87 -92 vol.1.

[29] Kovacs-Vajna, Z.M.; **"**A fingerprint verification system based on triangular matching and dynamic time warping", Pattern Analysis and Machine Intelligence, IEEE Transactions on , Volume: 22 , Issue: 11, Nov. 2000 Pages:1266 – 1276.

[30] Jain, A.K.; Lin Hong; Pankanti, S.; Bolle, R., "An identity-authentication system using fingerprints" Proceedings of the IEEE, Volume: 85 , Issue: 9 , Sept. 1997 Pages:1365 – 1388.

[31] J.G Daugman, "High confidence visual recognition of persons by a test of statistical independence."; Pattern Analysis and Machine Intelligence, IEEE Transactions on ,Volume: 15 , Issue: 11 , Nov. 1993 Pages:1148 – 1161.

[32] Li Ma; Tieniu Tan; Yunhong Wang; Dexin Zhang, "Efficient iris recognition by characterizing key local variations"; Image Processing, IEEE Transactions on , Volume: 13 , Issue: 6 , June 2004. Pages:739 – 750.

[33] Wildes, R.P.; Asmuth, J.C.; Green, G.L.; Hsu, S.C.; Kolczynski, R.J.; Matey, J.R., "A system for automated iris recognition" ; McBride, S.E.; Applications of Computer Vision, 1994., Proceedings of the Second IEEE Workshop on , 5-7 Dec. 1994. Pages:121 – 128.

[34] Boles, W.W.; Boashash, B., "A human identification technique using images of the iris and wavelet transform"; Signal Processing, IEEE Transactions on [see also Acoustics, Speech, and Signal Processing, IEEE Transactions on] ,Volume: 46 , Issue: 4 , April 1998 Pages:1185 – 1188.

[35] A. Ross and A. K. Jain, "Information Fusion in Biometrics", Pattern Recognition Letters, Special Issue on Multimodal Biometrics, Vol. 24, No. 13, pp. 2115-2125, September 2003.

[36] U. Dieckmann, P. Plankensteiner, T. Wagner, Sesam: A biometric person identification system using sensor fusion, Pattern Recognition Letters 18 (9) (1997) 827-833.

[37] J. Kittler, M. Hatef, R. P. Duin, J. G. Matas, On combining classifiers, IEEE Transactions on PAMI 20 (3) (1998) 226-239.

[38] E. Bigun, J. Bigun, B. Duc, S. Fischer, Expert conciliation for multimodal person authentication systems using Bayesian Statistics, in: First International Conference on AVBPA, Crans-Montana, Switzerland, 1997, pp. 291-300.

[39] A. K. Jain, L. Hong, Y. Kulkarni, A multimodal biometric system using fingerprint, face and speech, in: Second International Conference on AVBPA, Washington D.C., USA, 1999, pp. 182-187.

[40] P. Verlinde, G. Cholet, Comparing decision fusion paradigms using k-NN based classifiers, decision trees and logistic regression in a multi-modal identity verification application, in: Second International Conference on AVBPA, Washington D.C., USA, 1999, pp. 188-193.

[41] D.Gajski, N.Dutt, A.Wu, and S.Lin:, High-level Synthesis: introduction to Chip and System Design, Kluwer Academic Publishers, 1992.