

AFIM: A High Level Conceptual ATM Design Using Composite Formal Modelling With Capture Simulation Pattern Matching Technique

¹Okafor K .C, ²Udeze C.C, ³Ugwoke F.N, ⁴Obinna Ifesinachi ⁵Okwuelu Nnaemeka
^{1,2,5}Electronics Development Institute, Awka, Nigeria,

³Computer Science, Michael Okpara University Umuahia, Nigeria,

⁴Electronic and Computer Engineering, Nnamdi Azikiwe University Awka,

¹arissyncline@yahoo.com, ²udezechidiebele@yahoo.com, ³azubuikec@yahoo.com, ⁴ifinobi@yahoo.com, ⁵god4meks@gmail.com

Abstract—This paper is an extension of our earlier work on enhanced technique in ATM risk reduction using Automated Biometric Fingerprint in Nigeria. A conceptual model of an Automated Fingerprint Identification Machine (AFIM) is presented as an engineering solution to detailed AFIM designs for financial transactions. We present detailed Data Flow Diagrams (DFDs) and description algorithms as a Formal Modelling Framework (FMF). Our framework is a part of an ongoing research in improved Automated Teller Machines leveraging biometric fingerprint and cryptographic schemes. In this work, a pattern matching frame for an input image set at 0.99% threshold was realized with MATLAB Simulink for identification and verification purposes of the AFIM internal logic. Future work will show our GUI implementation results leveraging Software Development level Cycle activities.

Keywords: Algorithms, ATM, Conceptual, Cryptography, Fingerprint, Framework, Modelling, Transaction

INTRODUCTION

An Automated Teller Machine (ATM) is a computerized telecommunications device that enables the clients of any financial institution to perform financial transactions like deposits, transfers, balance enquiries, mini statement, withdrawal and fast cash etc., without the need for a cashier, human clerk or bank teller [1]. There are two types of ATMs: first, it is a simple ATM used only for cash withdrawal and to receive a report on accounts balance and second one is a complex unit, which is used for deposits and money transfer [1]. The former is the most frequently used in most countries of the world.

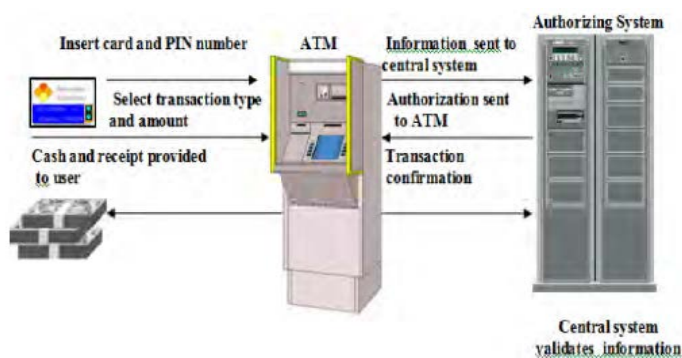


Fig.1: Conventional ATM model

On most modern ATM designs, the customer is identified by inserting a plastic ATM card with a magnetic stripe or a plastic smart card with a chip that contains a unique card number and some security information such as an expiration date. In this case, authentication is provided by the customer entering a personal identification number (PIN). Using this type of ATM model, customers can access their bank accounts in order to make cash withdrawals, debit card cash advances, and check their account balances as well

as purchase prepaid cell phone credit. The choice of ATM system determines the type of encryption modeling on the complete system.

The existing work in [2] focuses on Data Encryption Standard and Advanced Encryption Standard as the encryption standards used by the banks to protect the data and for secure data transmission. The work in [3] focused on ATM biometrics leveraging use case diagrams, activity diagrams with V.B.net implementation. The paper in [4] proposed a framework for credit cloning.

The authors in [5] proposed a method to improve the security Level of ATM Banking Systems Using AES Algorithm by introducing Encrypting PIN Pad (EPP) as shown in Fig. 2. The authors in [6], formulated a suitable simulation technique which will reduce idle time of servers and waiting time of customers for any bank having ATM facility, and argued that the technique will be helpful for any bank at global level for improving their customer's service towards competitive advantage. Fig. 2 shows a typical ATM with its Encrypting PIN Pad (EPP).



Fig.2.1: An ATM setup with EPP [5]

This paper observes that regardless of the cryptographic or biometric design of an ATM system, the role of formal modeling for such designs will reveal the engineering ingenuity more accurately. We now discuss on formal method philosophy as well as our contribution in this work.

FORMAL METHOD ENGINEERING

In software engineering, formal methods are a particular kind of mathematically based techniques for the specification, development and verification of software and hardware systems [7]. The use of formal methods for software and hardware design is motivated by the expectation that, as in other engineering disciplines, performing appropriate mathematical analysis can contribute to the reliability and robustness of a design [8]. Formal methods are best described as the application of a fairly broad variety of theoretical computer science fundamentals, in particular logic calculi, formal languages, automata theory, and program semantics, but also type systems and algebraic data types to problems in software [9]

According to [10] a real-time control system, the ATM system is characterized by its high degree of complexity, intricate interactions with hardware devices and users, and necessary requirements for domain knowledge. There is no systematic and detailed documentation of design knowledge including the modeling prototypes of ATM systems. Hence the role of denotation mathematics and formal notation systems will suffice. The author in [10] presented the formal design, specification, and modeling of the ATM system using a denotational mathematics known as Real-Time Process Algebra (RTPA). In this case, the RTPA used process relations to describe software system architectures as well as its behaviours with a stepwise refinement methodology. According to the RTPA methodology for system modeling and refinement, a software system can be specified as a set of architectural and operational components as well as their interactions. The use of Unified Data Models (UDMs), also known as the component logical model (CLM) which is an abstract model of system hardware interfaces, an internal logic model of hardware, and/or an internal control structure of the system could be leveraged. Also, the use of unified process model for modelling static and dynamic processes is vital in this regard.

RESEARCH CONTRIBUTIONS

We present a basis for developing an ATM system in a top-down approach using Conceptual Formal Design methodology. The architectural model of the ATM system is created based on the Real Time Process Algebra (RTPA) architectural framework. With the formal model of the ATM system, code can be automatically generated using V.B.Net framework. The models will serve as a formal design paradigm of real-time software systems as well as a test bench for the expressive modeling capability in software engineering generally.

Secondly, we demonstrate a MATLAB simulation for bio-

metric image processing using an ATM process matching controller while stating our assumptions.

METHODOLOGY

Composite Formal Modelling

A formal characterization of the ATM is realized using a mathematical concept known as Real-Time Process Algebra (RTPA). For the code generation, Object oriented design approach will be used in the Interface design using SDLC/V.B.Net in our future work.

Phases for Formal Methods

i. Specification Stage

We used composite formal method to give a description of the ATM system to be developed, whatever all level(s) of detail desired. This formal description was detailed to guide further development activities in the ATM design. This was used to verify that the requirements for the system after a complete and accurate specification. The specification of ATM model before its development is considered crucial as it serves as bench marks for evaluating design as well as its implementation. This also facilitates quality assurance via verification and validation.

ii. Development Stage

After the formal specification was produced, the specification now served as a guide while the ATM system is been developed incrementally during the design process. In this case, if formal specification was made in an operational semantics, the observed behavior of the concrete ATM system is then compared with the behavior of the Specification. Additionally, the operational commands of the specification may be amenable to direct translation into executable code. Also, the formal specifications, the preconditions and post conditions of the specification now become assertions in the executable code.

iii. Verification Stage

With the development of a formal specification, the specification is now used as the basis for proving properties of the ATM specification.

CONCEPTUAL AFIM FLOW DIAGRAMS

In our proposed system, the methodology of structured systems analysis & design provided a roadmap for the development of functional specifications for an intelligent system like the AFIM. In this work, an abstract and hierarchical ordering strategy used to manage the complexity of AFIM for design analysis was developed as shown in Fig. 3.3 [12]. In our design, the key functions of the intelligent system (AFIM) are centered on the AFIM dictionary which integrates the Data Flow Diagrams, Use Case Models, State Tran-

sition Diagrams, Control flow Diagrams and decomposition diagrams.

fications of the proposed system. These logical DFDs are also levelled and balanced.

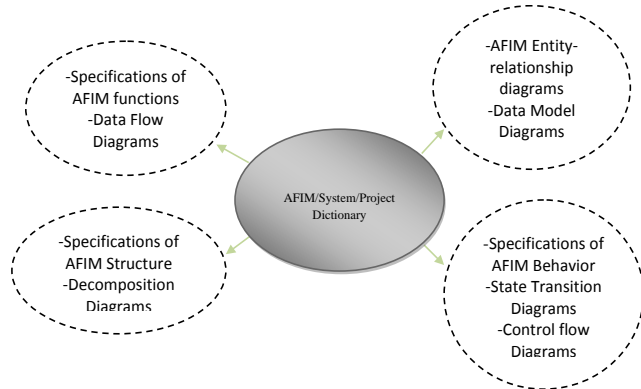


Fig.3.3: Hierarchical Ordering (AFIM Component Specifications)

The functional specifications are documented graphically in Dataflow Diagrams (DFDs) with a brief description below.

STEP 0: (Defining the scope of the existing system under study) - This accomplished by drawing the context diagram for the existing system.

STEP 1: (Documentation of how the existing system work) - This was accomplished by drawing the Physical Data Flow Diagrams. These DFDs specify the current implementation of existing system, and would answer questions such as:

- Who performs the tasks in existing system?
- How they are performed in existing system?
- When or how often they are performed in existing system?
- How the data is stored in existing system?
- The task process, is it secured in existing system?
- How is the data flows implemented in existing system?

These physical DFDs may be levelled, or, if the system is not very large, prepared all on a single DFD.

STEP 2: (Documentation of what the existing system does.)- This is documented in logical DFDs of existing system. Deriving these logical DFDs of an existing system from the physical DFDs involve abstraction of all implementation details as shown in the AFIM later in this section. From the formal design description, all the implementation details are abstracted from the user hence OOADM applied logically. These logical DFDs are levelled in order to reduce the perceived complexity of the system, and balanced in order to assure consistency in the design.

STEP 3: (Documentation of what the proposed system will do.) After step 2, an examination or performance evaluation is carried out to ascertain why the existing system (ATM) does not meet the user requirements, and how it can be modified in order to meet such needs. The result is a set of logical DFDs which describe what the modified (proposed AFIM) system will do. These functional specifications are devoid of implementation considerations, and therefore rather abstract speci-

STEP 4: (Documentation of how the proposed system will work.) The logical DFDs of the proposed AFIM system derived in step 3 above are then examined to determine which implementation of it meets the user requirements most efficiently. The result is a set of physical DFDs of the proposed system. They answer questions such as:

- Who will perform the various tasks in AFIM?
- How they will be performed in AFIM?
- When or how often they will be performed in AFIM?
- How the data will be stored in AFIM?
- The task process, is it secured in AFIM?
- How the data flows will be implemented in AFIM?

In this step, man-machine boundaries are drawn, and media selected for all data flows and data stores. This is well represented in chapter 4. Also, Fig. 3.6 shows the logical DFD based on level 1 formal method The function of any intelligent system can be expressed in terms of transformation (processing) of certain inputs (which are data) into outputs (which are data too) where memory (which too consists of data) may need to be consulted (or updated). This suggests that two crucial elements in a system's description are data and processing of data. A complete description of an intelligent system demands description of both these elements. In context, this work defines a system by this equation:

$$System (Afim_S) = Crypto_biometricData (CBd) + Processing\ of\ the\ data$$

While it is impossible to describe an intelligent system exclusively in terms of data or its processing, it is possible to study a system's functions (what the system must do) in terms of the transformations it must perform separately. An intelligent system's functions, which describe its processing aspects, is modeled in the structured systems approach, as dataflow diagrams. Such a model of an information data system is referred to as functional model or process model. A data-flow diagram in all cases consists of external entities, data stores and data-flows. Fig.3.4 shows a DFD framework, (DFD of structured systems analysis & design approach) but Fig.3.5 shows the AFIM abstract DFD implemented in Appendix I, II, III and IV. Fig. 3.6 shows the logical DFD based on level 2 formal methods.

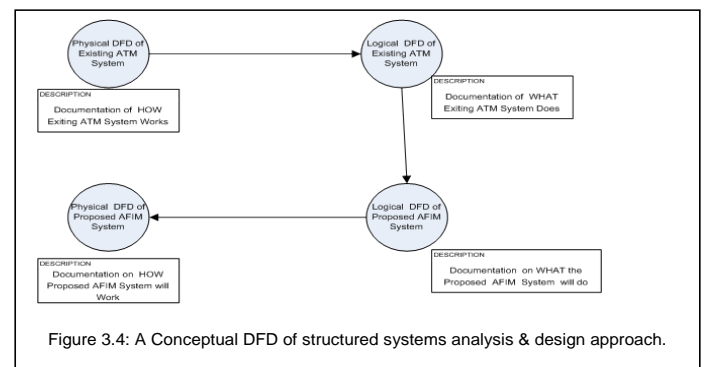


Figure 3.4: A Conceptual DFD of structured systems analysis & design approach.

In Fig. 3.5, with the DFD transaction errors could result from wrong pattern matching, insufficient balance, security pin errors and other contextual scenarios as discussed later in this

work.

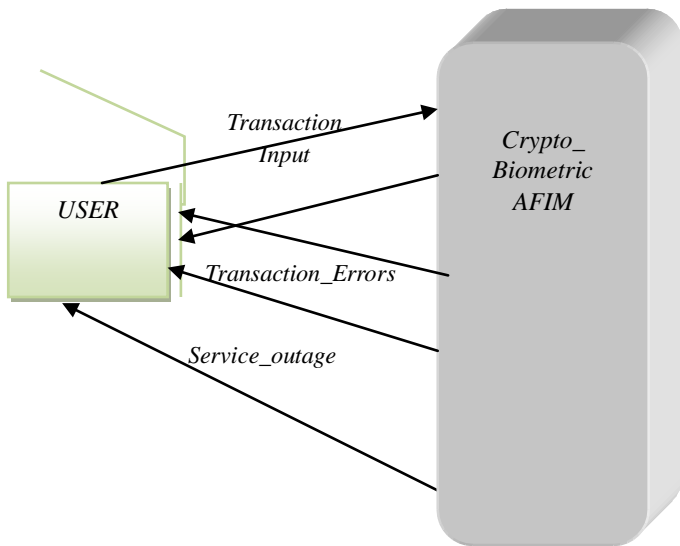


Fig.3.5: User/Actor Transaction DFD in AFIM [12]

BIOMETRIC FINGERPRINT IDENTIFICATION MACHINE MODEL

The proposed fingerprint pattern matching and verification algorithm flows is shown in Fig. 3.8. The data acquisition component is the first phase (phase 1 as presented below). The second phase comprise of the preprocessing steps. The third component employs a feature extraction algorithm to produce a feature vector whose best describe the characteristic of the fingerprint image notwithstanding the quality of the input image. The fourth component of the system generates the subject fingerprint image model. The last component compares feature vectors to produce a score which indicates the degree of similarity between the test fingerprint and subject fingerprint models.

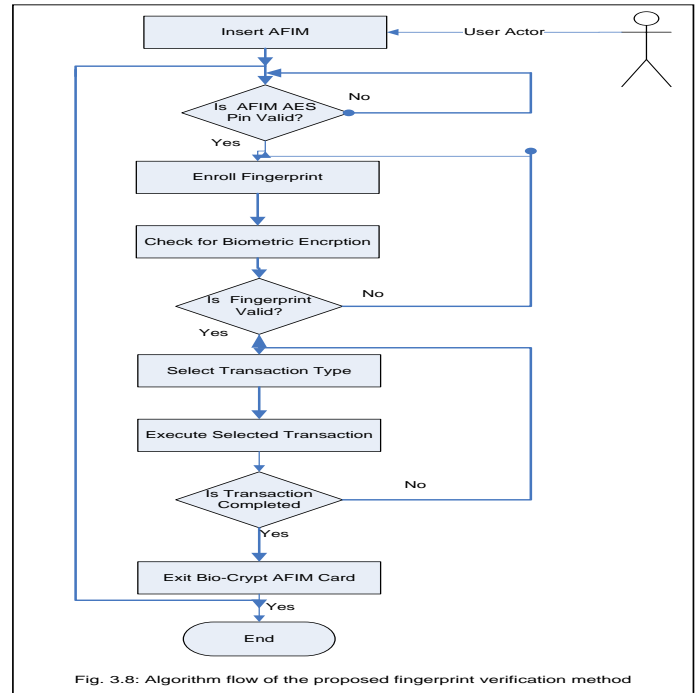


Fig. 3.8: Algorithm flow of the proposed fingerprint verification method

AFIM ARCHITECTURAL FRAMEWORK

In this work, the architecture of the AFIM is modeled as a real-time system. This model is a system framework that represents the overall structure, components, processes, and their interrelationships and interactions. This section gives the specification of the architectural model of the AFIM system denoted by AFIM Architecture_{ST} which is the high-level architectural framework based on the conceptual model developed in Fig. 4.1 [12]. From Fig. 4.1, the System architecture, at the top level, specify a list of structural identifiers of UDM and their relations. A UDM in context is a predefined class of the system hardware or internal control models, which can be inherited or implemented by corresponding UDM objects as specific instances in the succeeding architectural refinement processes for the system.

Modeling AFIM Using SDLC Real-Time Process Algebra (RTPA)

This work consequently models architectural component of an AFIM with discretional access control.

Now let,

AFIM_Card represent users' cryptographic AFIM card

AFIM_keypad represent the device input keypad

AFIM_Finger represent the AFIM finger input instance

AFIM_Monitor represent the display monitor

AFIM_Acct Database represents the user's database

AFIM_Billp represent the bill print out

AFIM_Clock represent the synchronization timing

We then developed the physical model in Fig. 4.1 as shown.

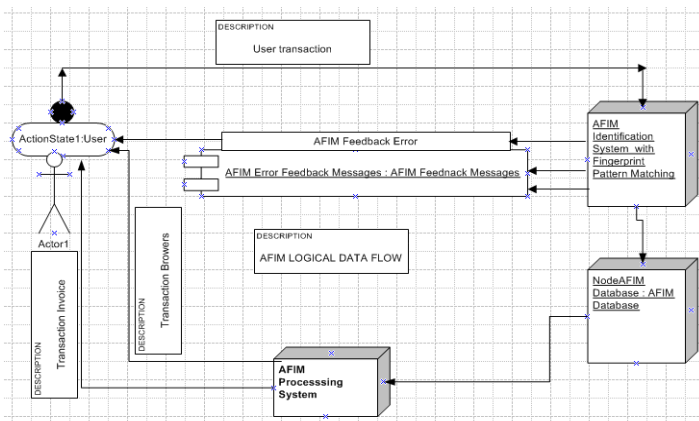


Fig.3.6: Logical DFD based on level-2 formal Method

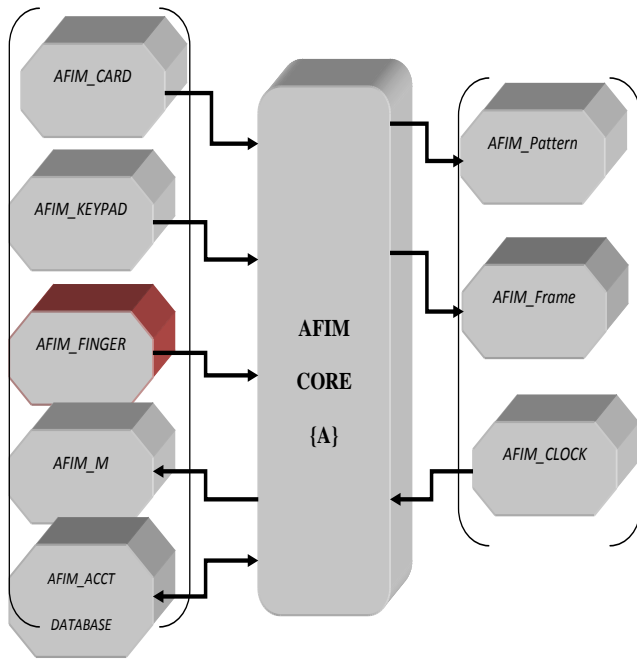


Fig.4.1: Physical model of the AFIM

From Fig. 4.1, the physical model of an AFIM is best described by a Finite State Machine (FSM), which adopts a set of states and a set of state transition functions modeled by a transition diagram or a transition table to describe the basic behaviors of the AFIM.

STATE FORMULATION FOR AFIM

Given an AFIM_core with which AFIM Finite State $\rightarrow (A, \Sigma, a, F, \delta)$

The objective function as explained in [12] would be:

$$\text{Max } \sum_{i=0}^{N=12} \{AFIM\} \text{ ----- } 1.0$$

Where $A \leq N$

- A is a set of valid states that forms the domain of the AFIM, $A = \{A0, A1, \dots, A8\}$ where the states are:
- A0 \rightarrow AFIM_System,
- A1 \rightarrow Welcome
- A2 \rightarrow Check PIN,
- A3 \rightarrow Biometric verification,
- A4 \rightarrow Input withdraw amount,
- A5 \rightarrow Seek Approval,
- A6 \rightarrow Verify balance,
- A7 \rightarrow Disburse bills, and
- A8 \rightarrow Eject card;

The Parameters of the existing Automated teller machine that the biometric is to be integrated upon is shown thus,

$$\sum_{i=1}^{N=12} AFIM \rightarrow \text{A set of events that the AFIM may accept and process, and}$$

$$\Sigma = \{i_0, i_1 \dots i_{12}\}$$

Where:

- $i_0 \rightarrow$ Start
- $i_1 \rightarrow$ Insert card
- $i_2 \rightarrow$ Correct PIN
- $i_3 \rightarrow$ Incorrect PIN,
- $i_4 \rightarrow$ Request \leq max,
- $i_5 \rightarrow$ Request $>$ max
- $i_6 \rightarrow$ Cancel transaction
- $i_7 \rightarrow$ Sufficient funds,
- $i_8 \rightarrow$ Insufficient funds,
- $i_9 \rightarrow$ Sufficient bills in AFIM,
- $i_{10} \rightarrow$ Insufficient bills in AFIM,
- $i_{11} \rightarrow$ Approval granted,
- $i_{12} \rightarrow$ Approval denied;

- A is the start state of the AFIM, $A = a_1$ (Welcome);
- F is a set of ending states, $F = \{a_1\}$;
- δ is the transition function of the AFIM that determines the next state of the FSM, s_{i+1} , on the basis of the current state and a specific incoming event i_v , i.e., $a_{i+1} = \delta(a_i, i_v)$, where $\delta = f: A \times \Sigma \rightarrow A$

Policy Algorithm

- Users insert AFIM, Card
- The user supplies PIN & Biometric data, seeking access
- PIN is verified and Biometric finger print is made & passed.
- If verification fails the user has the right to retry two more times.
- If at the third time PIN is confirmed BUT the fingerprint is wrong then there will be a mismatch on the AFIM and transaction is cancelled, Card ejected and AFIM is returned to welcome state.
- If PIN is correct & Biometric verification passes, then the user selects transaction details
- The machine verifies if amount requested is less than available account balance
- It also verifies if there are sufficient bills in the AFIM,
- Disburse bills; eject card and return to welcome state.

This policy explains the flowchart in Fig. 3.8. The conceptual model and descriptions of an abstract FSM model of the AFIM system is presented in the next section. The biometric enrollment found on the figure differentiates the proposed system from the existing system in that it is not usually found in other existing systems.

The top level framework of the AFIM can be modeled by a set of architecture, static behaviors, and dynamic behaviors using the class RTPA algorithm as follows for code generation:

```

Algorithm RTPA_I: §(AFIM)  $\rightarrow$  AFIM§.ArchitectureST
|| AFIM§.StaticBehaviorsPC
|| AFIM§.DynamicBehaviorsPC
    
```

Where || indicates that these three subsystems related in parallel, and §, ST, and PC are type suffixes of system, system structure, and process, respectively.

The conceptual models of AFIM as presented in Fig. 4.1

through 4.2, describes the configuration, basic behaviors, and logical relationships among components of the AFIM model. According to the conceptual system modeling, specification, and enumeration, the top level model of any system may be specified in a similar structure as given in equation 1.0. The following sections will extend and refine the top level framework of AFIM into detailed architectural models (UDM) and behavioral models.

As depicted in Fig. 4.1, a high-level specification of the architecture of AFIM, *ATMŞ.ArchitectureST*, is developed in Algorithm II, using RTPA. In this model, the *AFIMŞ.ArchitectureST* encompasses parallel structures of a set of UDMs such as the *AFIM_core: ST*,

- AFIM_CardReader: ST*,
- AFIM_Keypad: ST*,
- AFIM_Monitor: ST*,
- AFIM_BillStorage: ST*.

Algorithm RTPA II:

Architectural framework of the AFIM system {}

```

AFIMŞ.ArchitectureST :< AFIM_Core: ST |>
|| <AFIM_CardReader: ST |>
|| <AFIM_FingerPrint_Enroll: ST |>
|| <AFIM_Keypad: ST |>
|| <AFIM_Monitor: ST |>
|| <AFIM_BillStorage: ST |>
|| <AFIM_BillsDisburser: ST |>
|| <AFIM_SysClock: ST |>
|| <AFIM_SysDatabase: ST |>
|| <@EventsS>
|| <StatusBL>
    
```

The set of events of AFIM are predefined global control variables of the system, as given in equation 1.0, which represent an external stimulus to a system or the occurring of an internal change of status such as an action of users, an updating of the environment, and a change of the value of a control variable.

Algorithm RTPA III:

ATMŞ.ArchitectureST.EventsST →:

@*SysInitialS*

- | @*tTM* = \$*thh:mm:ss*
- | @*SysClock100msInt* int

We shall present the code generation with the GUI interfaces for the conceptual formal descriptions presented above. This work will now present the simulation capture for AFIM pattern matching identification scheme.

CAPTURE SIMULATION FOR PATTERN MATCHING

As stated in the previous section, this paper seeks to develop a biometric pattern matching model for fingerprint images using a threshold index of 0.99. The capture computer is realized with the MATLAB Simulink vision system toolbox [II]. This was used to model a 2-D normalized cross-correlation for pattern matching for user fingerprint inputs. The relative size of the target to the input image pattern and the pyramiding factor determine which domain computation is faster. In this work, the fingerprint recognition model compares the given input image with the template fingerprint image that is stored in the database. The matching processes comprises of four steps, viz:

1. Image acquisition-which is done by the optical multispectral sensor.
2. Image enhancement-used to adjust the contrast of the image and also to remove noise.
3. Fingerprint image segmentation- this is used to extract the region of interest, which is further used for feature extraction such as minutiae.
4. Matching-based on the extracted features, the verification of the given image is done with the image in the database and returns the result either as true or false.

The matching result determines whether the customer can access the account or not. From Fig. x1, the fingerprint image selected undergoes a rotation which generally commutes with the Fourier transform for continuous two-dimensional signals. The ROI on the image is monitored on 0.99 thresholds. The matching processor carries out the pattern matching algorithm above. In this case, the matching of two fingerprints is done based on the features extracted from the fingerprint. The AFIM uses minutiae based matching technique, since the minutiae are the most important features of fingerprint. This include ridge bifurcations- ridge dividing into two and ridge terminations- abrupt end of a ridge. Before the process of minutiae extraction, thinning of the segmented image is performed. Thinning is done, so that the ridges are only one pixel wide. After thinning, many spurious pixels may be formed, such as H-breaks and spikes. These spurs are removed using morphological operators. Its design is represented in the following diagram of Fig. x1 which shows the basic fingerprint pattern detection/identification process while Fig. x1 shows the Pattern Matching model:

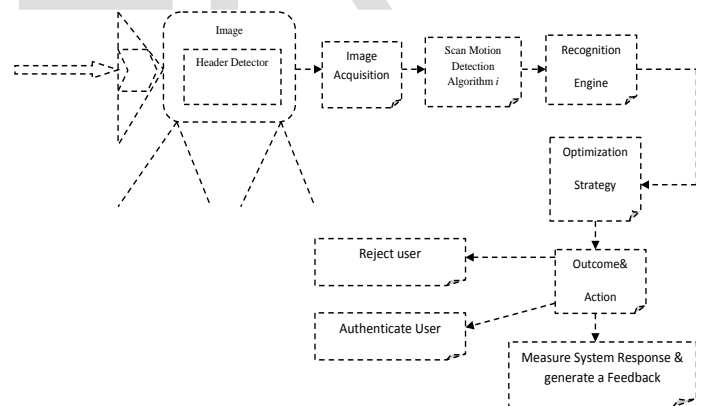


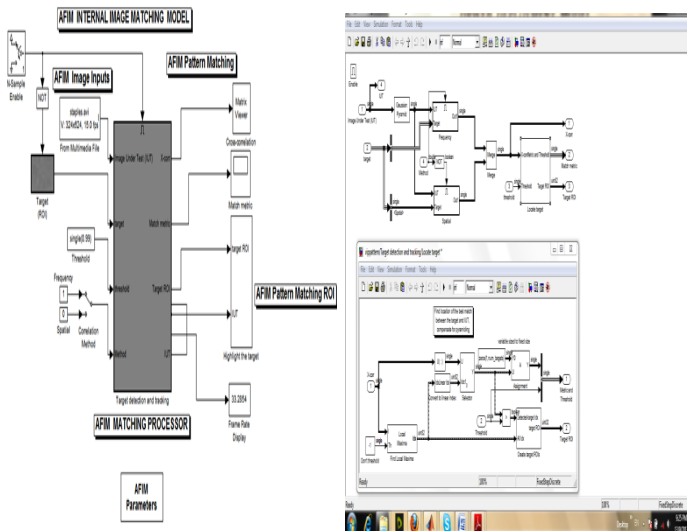
Fig.x1: Basic Fingerprint pattern detection/identification block diagram

Design Assumptions Formulations (Analytical Decompositions)

Some considered design specifications for the proposed MATLAB AFIM pattern matcher implemented with the computer human vision toolbox in MATLAB software, includes:

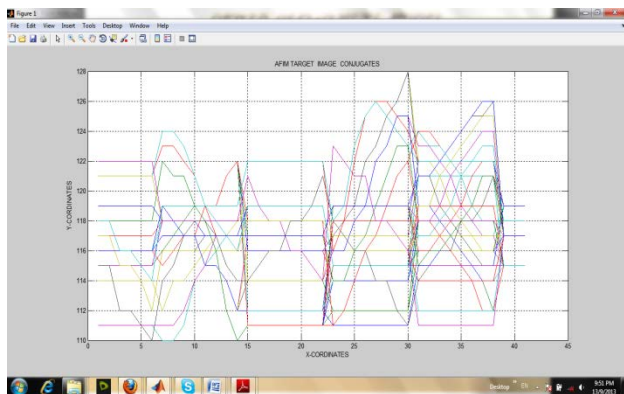
- i. A pre-set template is stored in the system and is called up by the input Data Acquisition System (DAS).
- ii. The Digital image processor will handle only single output data type (In context, one fingerprint image)

- iii. The Digital Signal processor used for the updating schemes is the Sum Absolute difference type
- iv. The image block sizes are vectors [60,80]
- v. The Threshold index is set at 0.99
- vi. The detection block is zero crossing.
- vii. The correlation pattern matching is 2-dimentional only



MATLAB Simulink AFIM Matching Processor Schematic Capture with matching processor subsystem

The match metric window shows the variation of the target match metrics. The model determines that the target template is present in a video frame when the match metric exceeds a threshold (cyan line). The Cross-correlation window shows the result of cross-correlating the target template with a video frame. Large values in this window correspond to the locations of the targets in the input image. The Overlay window shows the locations of the targets by highlighting them with rectangular regions of interest (ROIs). These ROIs are present only when the targets are detected in the video frame.



AFIM fingerprint image coordinates for Region of Interest

Conclusion

This paper has presented a conceptual formal modeling framework for a proposed AFIM. Data flow diagrams and

descriptive specification of AFIM is presented. A pattern matching model with MATLAB Simulink was realized while outlining the matching processes and procedures. The region of interest in our pattern matching model shows the most essential target of any biometric detection system. At 0.99, an excellent threshold index is observed. In the AFIM model, even if a user password pin ID is hacked, the model with the exact reference fingerprint allows only a valid finger image for the password ID. The proposed AFIM model offers better functionality and performance compared with other cryptographic and bio cryptosystem, but will be compared with biometric fuzzy faults model in our future work. Also the GUI results will be presented in our subsequent research works.

References

- [1] Sowmya.R., Sandhya, B.Thamotharan, S.Ramakrishnan, "A New Business Model For ATM Transaction Security Using Fingerprint Recognition, International Journal of Engineering and Technology (IJET), Vol 5 No 3 Jun-Jul 2013, pp. 2041-2047.
- [2] Navneet Sharma, Vijay Singh Rathore, "Different Data Encryption Methods Used in Secure Auto Teller Machine Transactions", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 - 8958, Volume-1, Issue-4, April 2012 pp.175-176.
- [3] Ibidapo, O. Akinyemi, Zaccheus O. Omogbadegun, and Olufemi M. Oyelami, "Towards Designing a Biometric Measure for Enhancing ATM Security in Nigeria E-Banking System. International Journal of Electrical & Computer Sciences IJECS-IJENS Vol: 10, No.6.
- [4] Divya Singh, Pratima Kushwaha, Priyanka Choubey, Abhishek Vaish*, and Utkarsh Goel, "A Proposed Framework to Prevent Financial Fraud through ATM Card Cloning" Proceedings of the World Congress on Engineering 2011 Vol I, WCE 2011, July 6 - 8, 2011, London, U.K.
- [5] N.Selvaraju, G.Sekar, "A Method to Improve the Security Level of ATM Banking Systems Using AES Algorithm" International Journal of Computer Applications (0975 - 8887) Volume 3 - No.6, June 2010.
- [6] Vasumathi.A , Dhanavanthan P, "Application of Simulation Technique in Queuing Model for ATM Facility" International Journal Of Applied Engineering Research, Dindigul, Volume 1, No 3, 2010. Pp.469-482.
- [7] R.W.Butler (2001-08-06). "What is Formal Methods?". Retrieved 2006-11-16.
- [8] C. Michael Holloway. *Why Engineers Should Consider Formal Methods*. 16th Digital Avionics Systems Conference (27-30 October 1997). Retrieved 2006-11-16.
- [9] <http://en.wikipedia.org/formal> development
- [10] Yingxu Wang, "The Formal Design Model of an Automatic Teller Machine (ATM)
- [11] www.MathWorks, Inc., 7.12 (2011b)
- [12] Obinna et al. "An Enhanced Technique in ATM Risk Reduction Using Automated Biometrics Fingerprint in Nigeria". Accepted for publication in International Journal of Science and Engineering.