

# Design Patterns and Object-Oriented Models of a Civil Biometric Service System

Abdelkader Mazouz

Neoma Business School, Mont Saint Aignan, France

C. Jim Han, Jatni Blandon

Florida Atlantic University, Florida, USA

The number of background and security checks in civil applications has increased dramatically in recent years. Most of those civil applications use the local and federal law enforcement agencies' databases. Because of the unpredictable civil application requirements and the development of new technologies, the biometric industry is apposite to use the design patterns and object-oriented analysis in developing and managing such dynamic and complex systems. This article presents the use of the design patterns as a common way to organize objects to make practical design decisions helping to generate flexible, manageable, and agile biometric software architectures. First, a conceptual model is presented modeling an Electronic Fingerprint Service (EFS) for civil applications based on object-oriented design. Then the model is demonstrated by applying design patterns and shows that the system can gain the flexibility and agility to expend and adapt to new biometric technologies and applications in an Electronic Biometric Service (EBS) system. Two cases are used to demonstrate the proposed method. This analysis and design interaction leads to the application of the design patterns by reusing the object created in the EFS model. The two case studies demonstrate how design patterns applied to the EBS model and show they can achieve flexibility and object reusability (agility) while maintaining the integrity and security of the main system model.

*Keywords:* Electronic Fingerprint Service (EFS), object-oriented, biometric service system, design patterns, biometric collection, fingerprint enrollment

## Introduction

Today's uses of biometric identification services focus on the ability to recognize individuals based on their physiological characteristics. The types of biometric information that can be obtained from a person range from DNA, to simple fingerprints, hand prints, retina, or voice. These identifiers are more reliable, secure, and efficient than traditional identification methods such as signatures or facial photos. The collection of biometric information is becoming more common in our daily lives. For instance, some banks require fingerprints when

---

Abdelkader Mazouz, MSE, Ph.D, Professor/MBB, Informations de Systemes, Supply Chain Management et Aide a la Decision, Neoma Business School.

C. Jim Han, Ph.D., Professor/ITOM, Department of Information Technology and Operations Management, College of Business, Florida Atlantic University.

Jatni Blandon, Researcher/MSE, Department of Ocean and Mechanical Engineering, College of Engineering, Florida Atlantic University.

Correspondence concerning this article should be addressed to Abdelkader Mazouz, 1 Rue Marechal Juin, Mont Saint Aignan, 76825, France. E-mail: Abdelkader-mazouz@neoma-bs.fr.

making banking transactions. Government buildings and other secure facilities use fingerprint authentication to provide general access. Pre-employment fingerprint background checks are required in order to apply for certain jobs. The biometric industry changes constantly with new technologies and new applications. Today's biometrics systems should be designed to handle continuous changes and improvements of the fast growing industry. Biometric systems are to be flexible, accurate, agile, and secure. Software development techniques such as object-oriented design and design patterns help create biometric service systems that can handle current and future application requirements.

### **Fast Growing Electronic Biometric Identification Technology**

Fingerprint biometrics services industry is one of the fastest growing industries today. The fingerprint is considered to be the most commonly used technique of all biometric identification systems. The first central database of electronic fingerprints—the Integrated Automated Fingerprint Identification System (IAFIS) was established by the Federal Bureau of Investigation (FBI) and other law enforcement agencies in July 1999 to unite the efforts to improve the fingerprint identification process (IAFIS, 2010).

The IAFIS system is a national fingerprint and criminal history system maintained by the FBI's Division of Criminal Justice Information Services (CJIS) (1999). The FBI-IAFIS is the largest biometric database in the world, with more than 47 million criminal records on file and is growing daily. With the proliferation commercially available electronic fingerprint scanners and the demand for fingerprint check in the civil/commercial areas, the FBI expanded its IAFIS services to provide both criminal and civil checks. The fingerprint background check process starts when the individual's fingerprints are obtained as a result of an arrest (criminal) or obtained due to a background check required for employment, licensing, and other non-criminal purposes (civil). The fingerprints are, in most cases, first processed by local authorities (state) or channeling agencies (agencies granted authorization to submit to the FBI) and then electronically forwarded through the CJIS Wide Area Network (WAN) to the FBI-IAFIS for processing. The FBI asserts that responses are returned in less than two hours for criminal electronic submission or less than 24 hours for civil electronic submissions. The FBI-IAFIS offers five different types of services which process 10-print based fingerprint, latent fingerprint, and other document and images.

### **Civil Usages of Electronic Biometric Services (EBS)**

Civil usages of EBS range from simple fingerprint matching to a whole criminal background checking which uses other biometric information such as hand scanners, electronic signatures, pictures (mug shots), voice, retina image, and others. In such applications, it requires the system to improve the ability to efficiently support submissions of various biometric devices and related applications to multiple agencies.

The system also needs to maintain the highest level of security. The biographic information obtained from the individual to be fingerprinted can contain other personal information such as social security numbers, date of birth, and other sensitive information together with the fingerprint data. It could represent an enormous problem if this information was to be misused in one way or another. Finally, the EBS system should interface efficiently with other supporting devices such as fingerprint card scanners, badge creation, fingerprint access control systems, and other application setups.

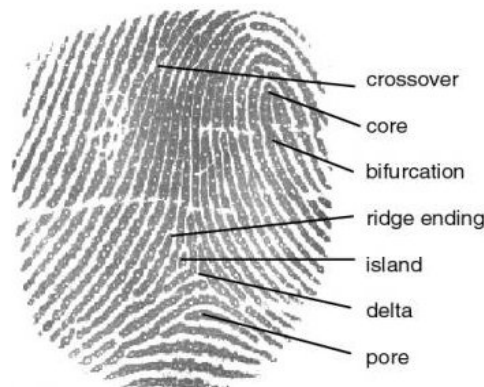
This paper demonstrates the use of design patterns as common way to organize objects to make practical design decisions that generate flexible, manageable, and agile software architectures. First we show the motive to this study. The background information about biometric fingerprint systems as an essential biometric

technology is then presented. The usage of object-oriented analysis and pattern design to develop flexible and agile systems are discussed in detail. Two design models using object-oriented design approach in an Electronic Fingerprint Service (EFS) system and using design patterns in an EBS system are presented as case studies. The case studies demonstrate how design patterns are applied to the EBS system model to achieve three primary goals: flexibility in application requirement management, objects reusability (agility), and maintainability in the accuracy of the main system model.

### **Biometric Fingerprint Technology and Identity Verification**

This section provides the background information on biometric fingerprint systems as an essential biometric technology. It describes how fingerprints are unique identifiers of a person's identity, and how the verification of a person's background is conducted by government agencies. It describes the different specifications and standards used to classify process and transmit biometric fingerprint information. It also describes current industry applications and provides a view into the factors affecting the continuous improvements and changes in Biometric Industry.

The fingerprint identity verification technology is at the forefront of the biometric identification systems. The extensive FBI master fingerprint database, fingerprint based access control systems, passports with embedded fingerprint technology are just a few of the major applications of this technology. Fingerprints are one of the most reliable types of biometric identification existing today. A CNN poll indicated that fingerprints rated high in social acceptance among other commonly known biometrics (Davide, 2003). A person's fingerprint minutiae characteristics (see Figure 1) are unique and permanent (Moenssens, 2006).



*Figure 1.* Fingerprint minutiae characteristics.

Fingerprint systems match finger image characteristics, not persons. There are two types of fingerprint recognition: verification and identification processes. The verification system authenticates a person's identity by comparing the fingerprint images with template stored previously in the system with the same name or social security numbers. The identification system identifies an individual by searching the entire template database for a match (Koraminski, 2005).

#### **Fingerprint Identification Process—Records and Devices**

The fingerprint verification process starts when the individual's fingerprints are obtained as a result of an arrest (criminal) or obtained due to a background check required for employment, licensing, or other non-criminal purposes (civil). The fingerprint record is, in most cases, first processed by local authorities (state),

and then forwarded electronically through the CJIS-WAN to the FBI-IAFIS for criminal records reprocessing. The fingerprint record needs to meet the standards dictated by the FBI, ANSI/NIST specifications (NIST, 2000). The fingerprint images can be captured electronically by using a FBI-certified live scan device or manually by inking and rolling the individual's fingerprints on a card (fingerprint card).

A rigid set of specifications must be followed by all fingerprint scanner systems and printers that will supply fingerprint data to the IAFIS, and to printers and displays supported by the IAFIS. They provide objective criteria for insuring image quality.

### **ANSI/NIST Data Standards and Specifications**

The data format for the interchange of fingerprint, facial, and Scar Mark & Tattoo (SMT) information standard defines the content, format, and units of measurement for the exchange of fingerprint, palm print, facial, and SMT image information that may be used in the identification process of a subject (NIST, 2000). This standard allows all AFIS and related systems to communicate among each other, to the FBI-IAFIS, and receive information from IAFIS. All transactions and messages to and from an IAFIS meet the ANSI standard for exchange of fingerprint information. The ANSI-NIST standard specifies the conventions for transmitting data (byte and bit ordering, gray scale data, binary data, color data, scan sequence, and etc.), image resolution requirements, scanner resolution, transmitting resolution, and file descriptions of the standard composition of the transmitted file.

The Common Biometric Exchange File Format (CBEFF) (NIST, 2001), describes a set of data elements necessary to support biometric technologies in a common way. These data can be placed in a single file used to exchange biometric information among different system components or among systems. The Information Technology Laboratory of NIST and the Biometric Consortium sponsored a workshop to discuss the potential of a "technology-blind" biometric file format that would facilitate the handling of different biometric types, versions, and biometric data. The resulting standard CBEFF was first published on January 3, 2001 (NIST, 2001). Smart cards (Di Maio, 2002) and registered traveler program (RTP) (Larman, 2002) are two examples of the applications based on the CBEFF standards.

This section covers general information about biometric fingerprint systems. We also outline the standards and requirements governing the creation, processing, and transmission of fingerprint records and other biometric information. Current applications of biometric systems and the factors driving the industry changes were described.

### **Robust Biometric Service System Using Object-Oriented Design and Design Patterns**

This section provides the view into the factors gearing the improvements and changes of Biometric Industry. It describes how the use of object-oriented analysis and design patterns can lead to the creation of robust, flexible, and extendable systems. The biometric recognition industry changes constantly with new technologies and new applications. Today's biometrics systems should be designed to handle the constant changes and improvements of this fast growing industry. Based on the current and near future civil biometric applications one can infer that a robust biometric system design should be flexible, accurate, secure, and fast. Those factors are the drivers for continuous changes and improvements in the biometric industry.

Flexibility is an important factor in design robust biometric system for civil usages. It is necessary to accommodate new requirements without a massive system redesign. It should handle different data types and

formats according to different application requirements. It should also interact seamlessly with other system architectures, hardware devices, and software application systems. However current IAFIS systems do not have the type of flexibility required to handle civil applications. Even systems at the federal level are not easy compatible with each other (Komarinski, 2005).

Accuracy is another important factor of a biometric system. A good biometric system should accurately determine a person's identity within minimum margin of error. It should be able to interoperate and combine multiple devices or different kinds of biometrics in order to determine or confirm a person's identity.

Timeliness is important in today's fast-passed world. Biometric systems are required to provide identification responses in a fast manner without compromising the accuracy of the results.

Finally, security is a very sensitive factor. Personal data must always be secure at all times, on servers, databases, smartcards, and over network transmissions. Biometric systems as any other system containing confidential personal information of an individual must be maintained current with the latest security assessments and technology.

The biometric identification technology changes with the new computer software, hardware, and network technologies. It continuously adds new technologies to its applications. From the civil applications perspective, we constantly find new applications using those existing local and federal biometric systems. It is therefore essential to consider the flexibility, security, searching speed, and accuracy in the system design stage. Object-oriented design and design patterns provided such a tool.

### **Object-Oriented Design and Design Patterns**

Object-oriented analysis and design are essential in the creation of well-designed, robust, and maintainable software (Larman, 2002). Object-oriented analysis emphasizes on finding and describing the object or concept in the main problem. Object-oriented design focuses on defining software objects and how they collaborate to fulfill the functional requirements of the software system. Object-oriented concepts are considered important in software reuse and evolution because they address fundamental issues of software adaptation and evolution. Object-oriented methods are based on the concepts of encapsulation of classes and inheritance. Encapsulation can lead to systems that are more maintainable and inheritance can provide an approach for adapting a class in a systematic way. Unified Modeling Language (UML) is a language for specifying, visualizing, constructing, and documenting the artifacts of software systems, business modeling, and other non-software systems (UML, 2009). Especially, the use case, sequence, as well as class diagrams are common tools to describe the static and dynamic specifications of a large system, such as our EBS system. All UML diagrams included in this paper use the standard UML 2.0 notations.

Software reuse has been a goal in software engineering for years (Gomaa, 2004). In traditional software applications reuse is to have a library of reusable components and of an approach for indexing, locating, and distinguishing among similar components. Instead of reusing an individual software component, it is much more advantageous to reuse a whole design or subsystem, consisting of the components and their interconnections. This means reuse of the control structure of the application. Architecture reuse has much greater potential than component reuse because it is the large-grained reuse, which focuses on reuse of requirements and design.

Variability at the design level can be achieved by using design patterns. In general, a pattern in software development process describes a situation that a core problem and its variations frequently occur in design and implementation. Different design patterns can be used to model the solution to that problem in such a way that it

can be reused as a solution for all problems of that type, or pattern. Patterns help to find common, effective methods for achieving and solving problems. Design patterns describe a recurring design problem to be solved, a solution to the problem, and the context in which that solution works (Yacoub & Ammar, 2004). Design patterns are also referred as micro-architecture because it involves more than one class and the interconnection among objects from different classes. Different patterns are used in various stages of a software development cycle. Those patterns include design patterns, architectural patterns, analysis patterns, product line specific patterns, and idioms.

Used properly the object-oriented design and design patterns can greatly assist the design and implementation of a robust and secure system. We explore the advantages of applying those technologies in developing an EBS system in the next section.

### Biometric Service System Design Model

A conceptual model representing an EFS system using object-oriented analysis and design is discussed in this section. The analysis and design approach for the EFS system model consists of defining the use cases and use case steps, creating sequence diagrams to provide a dynamic model of the system objects, and constructing the class diagrams to describe the system static behavior. The EFS system model builds the picture of how the pieces should interact in order to handle the requirements of an ever changing EFS system.

A reusable, flexible system design should start by gathering common requirements across existing biometric applications and expand on the differences. We first discuss the system requirements for an EFS system and then expand it to an EBS system with additional applications.

### EFS System Analysis and Design

The EFS system being studied here is a basic fingerprint collection and processing system. It consists of three logical parts: customer enrollment, fingerprint capture, and submission to FBI/CAT. The process flow (see Figure 2) shows that the process starts when the customer selects the fingerprint service via a user interface. The system determines the data set the customer would need to provide in order to fulfill the request. The system enables the fingerprint capturing device and it processes the fingerprint images according to the respective quality standards. The system generates the EFT file (the file format specified by the FBI for fingerprint data submission) and then submits the fingerprint file to the FBI according to the specifications provided.

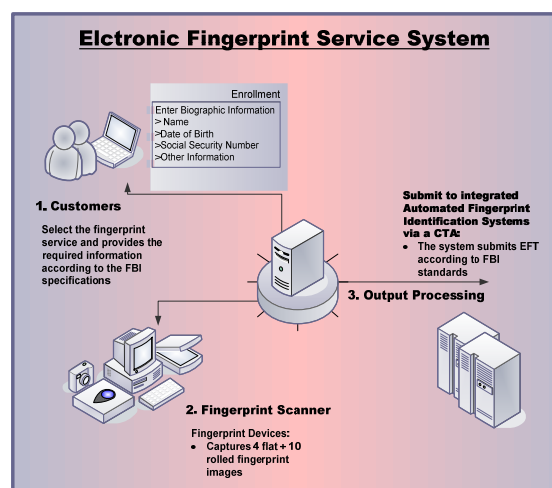


Figure 2. EFSS process flow.

### EFSS System Requirement Analysis (Use Case Model)

A high level analysis of the EFS requirements created the use cases (see Figure 3) diagram, which represents the high level description of the major functions/actors of the system. The system has four kernel use cases, which include account access, services, biometrics collection, and output management.

The account management use case relies on the account system actor to provide the account access and management functionality. It provides the customer with the ability to create, update, and close an account. It also validates the login information provided by a customer. Services management uses two main system actors, payment and fingerprint enrollment. Payment gathers and displays the payment options, and also processes payment information provided by the customer. Fingerprint enrollment gathers appropriate enrollment data fields, processes the biographic information provided by the customer, and then also validates and creates the biographic file record. Biometric collection uses the fingerprint system actor to capture and process the fingerprint information provided by the customer, and also validates the fingerprint quality according to the specified standards. Output manager uses the output system actor to process the transaction output, create EFT file, and processes the submission to the FBI/CTA.

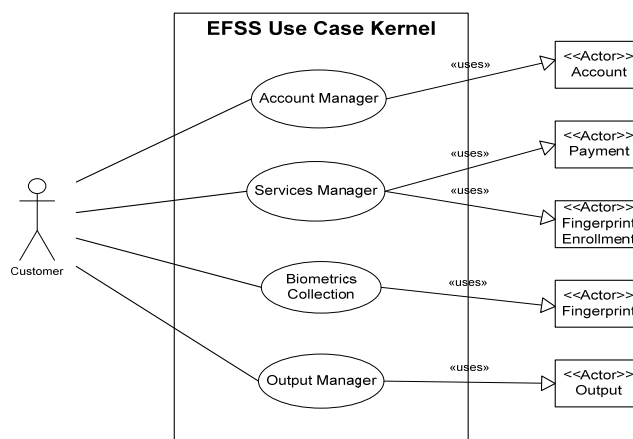


Figure 3. Biometric system use case diagram.

### EFSS Sequence Diagrams (Dynamic Model)

The dynamic models describe the objects and their interactions in the system. Sequence diagram is one of the most common UML diagrams for this purpose. There is a sequence diagram for each one of the use case shown in Figure 3. They represent the objects, messages, and the handling of the information among the objects.

The service sequence diagram in Figure 4 shows the service use case realization. The services object requests that the payment object for the payment options is displayed to the customer. The payment object processes the payment information sent by the customer, after which the payment object communicates with the fingerprint enrollment object. The fingerprint enrollment object gathers the appropriate enrollment data set for the service requested. After the customer provides the biographic information requested in the enrollment data set, the fingerprint enrollment object validates the data and creates biographic data file.

An account access sequences diagram to show the account access use case, a biometric collection sequences diagram to show the biometric collection use case, and an output manager sequences diagram to show that the output manager use case can all be realized use similar notation as shown in Figure 4.

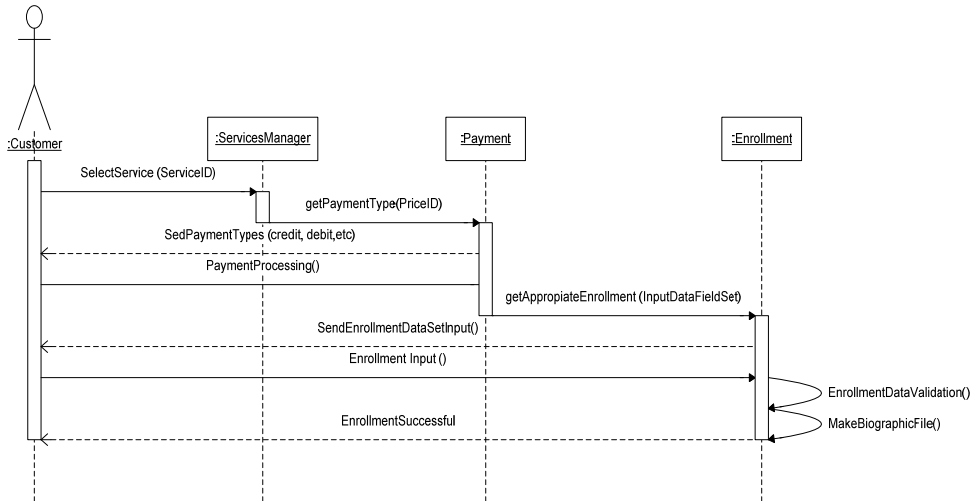


Figure 4. Services sequences diagram.

**EFSS Class Diagram (Static Model)**

The static model describes the conceptual system classes and the relationships among them. The EFS class diagram is presented in Figure 5. It is a conceptual class diagram because most of the classes do not contain the appropriate attribute types and their corresponding units. Since attribute types and units are sometimes specific to a particular programming language, this conceptual model is flexible and independent from a specific implementation. This class diagram was created using UML notations for inheritance, associations, and abstraction.

There are two major classes depicted in this class diagram. The account class handles the customer account logging and access. The services class contains the payment, fingerprint enrollment, biometric collection, and output manager subclasses. It controls the access and flow of most of the activities. The account class also uses the services class.

The payment class has two specializations (interfaces): credit card and PayPal. The fingerprint enrollment class is associated with the IAFIS enrollment specifications class. This association extends to the biographic record class. The instantiation of these classes produces a valid fingerprint enrollment and biographic record according to the IAFIS specifications. The biometric collection class is an abstract class supporting the fingerprint subclass. The fingerprint subclass has an association with the fingerprint specifications class for quality verifications and the fingerprint record class. It also has a specialization (interface) with the Xscan fingerprint scanner class. The instantiation of these classes produce a valid fingerprint biometric record according to the IAFIS specifications. The output manager class is an abstract class supporting the output subclass. The output class has two subclasses, the IAFIS logical record class and the IAFIS CTA submission class. These classes have subclasses to support their corresponding specifications, either following the IAFIS EFT specifications or the IAFIS-CTA submission specifications. The instantiation of these classes creates a valid EFT file and submits it to the specified agency.

This section describes the EFS system design. This model builds a picture of how the pieces interact in order to leverage future requirements. The system was modeled using object-oriented analysis and the design was defined by use case diagram, sequence diagram, and class diagram. UML 2.0 notation was used to create these diagrams. It outlines the importance of creating the analysis and design steps as a precursor to the code



implementation, as it provides the information desired for future improvements. Object-oriented design fundamentally addresses issues dealing with the reusability of objects and flexibility of the system.

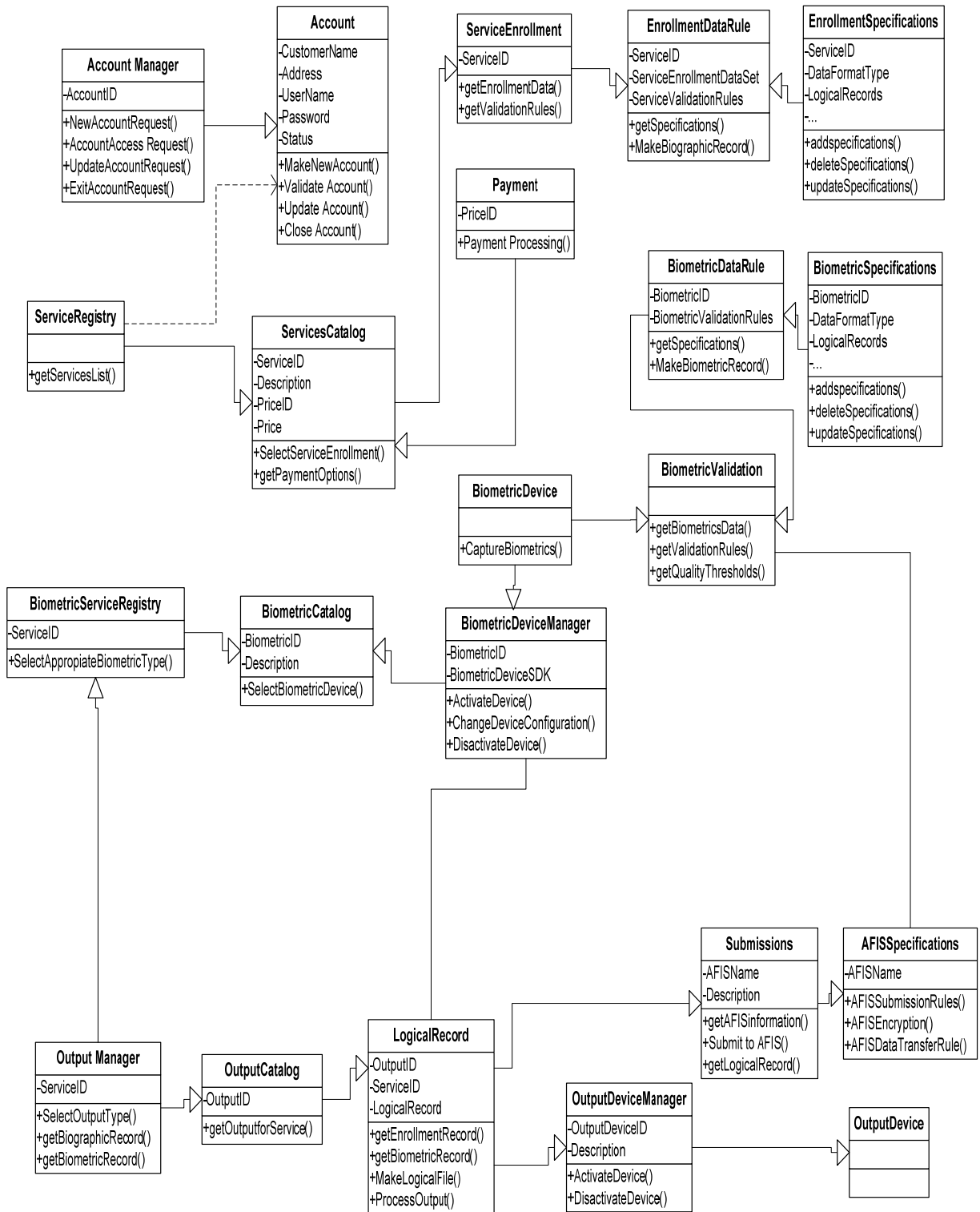


Figure 5. Class diagram-comprehensive biometric service system.

## Biometric Service System Case Studies

### Advantages of Using Object Oriented (OO) and Design Pattern in Developing EBS System

This section demonstrates that the original EFS model could be more flexible and agile to handle a change in requirements. A more elegant solution is needed to handle changes with flexibility, reusability, and maintainability. Some of the new requirements and changes to the current EFS system might occur as described below:

- Handling new requirements, such as new services and new enrollment data sets;
- Handling a new biometric device and specifications, such as voice recognition;
- Handling different types of output devices or connections, such as fingerprint card scanners, and smart card writer/reader;
- The design needs to handle changes on the data formatting using to generate output logical record (EFT or CBEFF).

Design patterns can be applied to the EFS system model to achieve requirement management flexibility, object reusability (agility), and manageability while maintaining the accuracy of the system.

In order to demonstrate this premise, two case studies were created. The first case study uses the EFS system as a premise to verify how it handles a new set of requirements, such as the addition of a new biometric device and service (a voice recognition device). Even though, it is a manageable set of requirements, which requires a new object-oriented analysis and design cycle to determine the points of variability and similarity of the two systems. This case study shows how a small change in the requirements can increase the number of objects in the original system limiting its ability to handle further changes, and therefore limiting its overall agility.

The second case study requires the addition of a new service that is to support the frequent traveler registration program (Larman, 2002). The service also requires the support of a voice recognition device. The EBS system requirements are described, and this set of requirements is more complex than that of previous models. The new EBS model is created using pattern oriented analysis and design which leverages the re-use of objects determined by the previous case study model. The EBS model is analyzed in order to select the design patterns that would simplify the handling of this more complex set of requirements in a more efficient manner. Design patterns provide a higher level of flexibility and reusability of objects than the object-oriented approach because they address common design problems in a much more abstracted level. Also, as building blocks, they replace objects or reuse objects therefore simplifying the system design. Design patterns allow the system model to become more flexible and agile in responding to future changes. The outcome of the pattern analysis process indicates that three main design patterns can be used in our EBS system. Those three design patterns demonstrated here are abstract factory, builder, and strategy patterns.

### Case One: EFS System With Voice Recognition Support

It is required that the current EFS model support a voice recognition device. A new object-oriented analysis and design cycle is required in order to determine the points of variability from the original model. The design of this new fingerprint service system consists of the following logical parts: customer enrollment, fingerprint-biometric capture, submission to FBI/CTA, and storage of the WAV voice file for output process. Figure 6 shows the process flow of the voice recognition system. The customer requests access to the account to be fingerprinted for a background check and adds the voice recognition file to the FBI-IAFIS for storage.

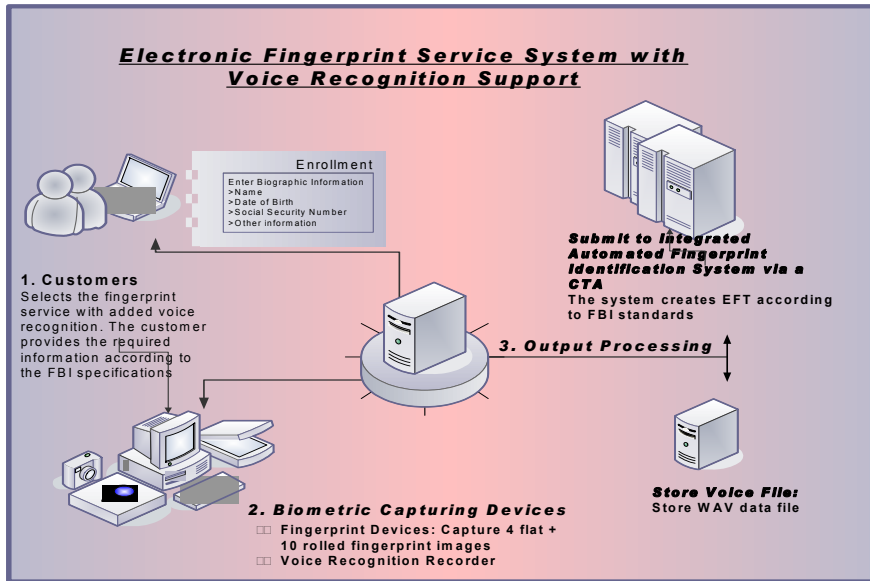


Figure 6. EFSS process flow with voice recognition.

A new use case diagram (see Figure 7) is created to indicate that two more system actors are required to support the new voice recognition device.

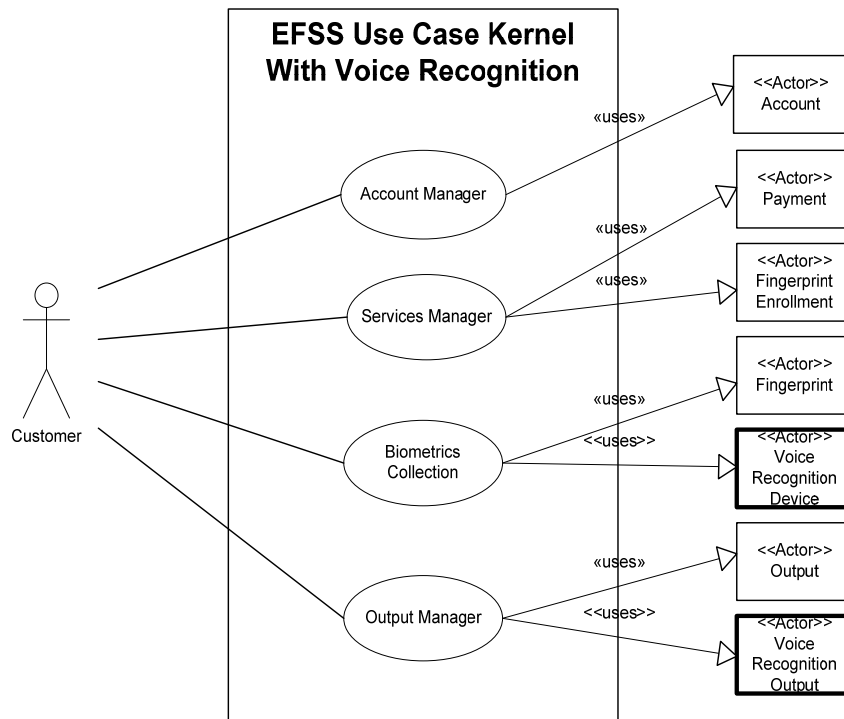


Figure 7. EFSS use case diagram with voice recognition support.

The new EFS class diagram on Figure 8 shows that in order to support the voice recognition device the system model requires the addition of a few new classes represented in bold boxes. These classes are to support the voice recognition specifications, quality, security rules, and device manager. These classes are different from those created to support the fingerprinting device.

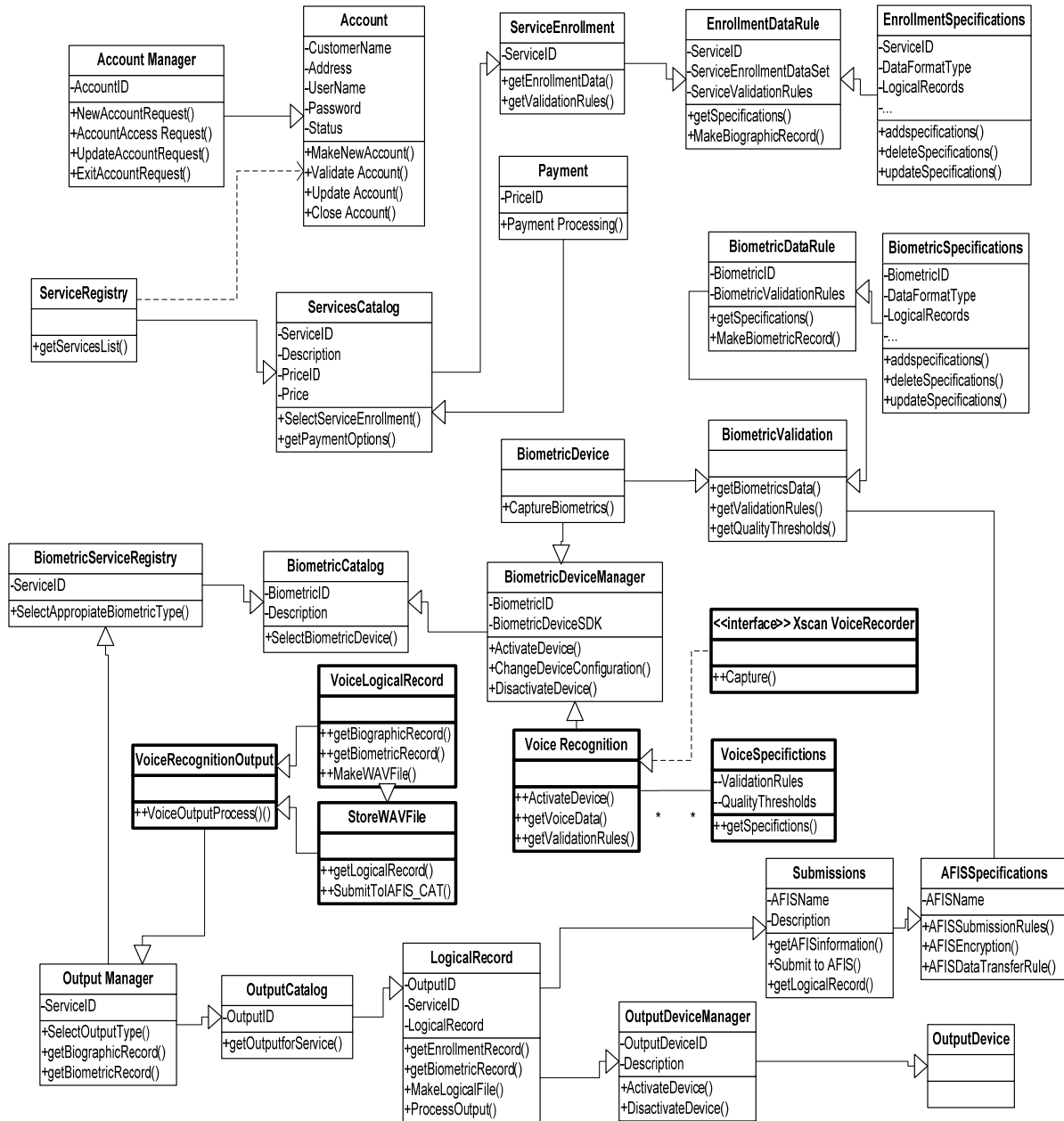


Figure 8. EFSS class diagram supporting voice recognition.

The biometric collection abstract class was extended to support the voice recognition capturing device and validation rules. Four new classes were added: voice recognition, voice record, voice specifications, and Xscan voice recorder (interface). The instantiation of these classes captures a valid voice biometric data. The output manager abstract class was extended to support the voice recognition output which consists of creating the WAV record and storing it according to the rules in the systems database. This requires five new classes, voice recognition output, voice logical record, voice specs, StoreWABFile, and WAV storage specs.

In order to apply this new set of requirements, new objects needed to be created. There was minimal reuse of the current object model. Its design model is manageable because the small set of requirements was determined to contain only few points of variability. This case study shows how a small change in the

requirements can increase the number of objects in the original system limiting its ability to handle other further changes and therefore limiting its overall agility.

### Case Two: RTP

The design of this biometric service system for RTP needs to support all common security check requirements and use technologies across existing biometric applications. The system could be considered as a kiosk containing different biometric services. Each service requires different set of input data and provides a set of outputs according to the requirements. For example, a service could consist of fingerprints with submission to a state AFIS, or fingerprints with submission to the FBI-IAFIS. Services could also include voice recognition, or smart card processing.

Figure 9 describes the process flow of the EBS system. It starts when the customer selects a service via a user interface. The system determines the input data set required to fulfill the request. The system selects the biometric capturing device or devices along with their respective quality and security rules and standards. One or more biometric types can be associated with a particular service request. The system also selects the types of output associated with the service selected by the customer. Once the customer provides the biographic data, the system prompts the customer to provide the biometric data. After the biometric capture process is successfully completed, the system prepares the appropriate output or outputs. The customer is given a final confirmation or result status according to the service selected.

The use case diagram in Figure 10 indicates that two more system actors are added to the EFS's original model to support the RTP. Those two new actors are the registered traveler service and the smart card output.

The abstract factory design pattern allows for creating families of related classes that implement a common interface. Figure 11 describes how, by using an abstract factory, both fingerprint and voice recognition biometrics can be accessed by one main abstract class (biometric abstract factory). This design pattern was selected to handle the design of multiple types of biometric devices.

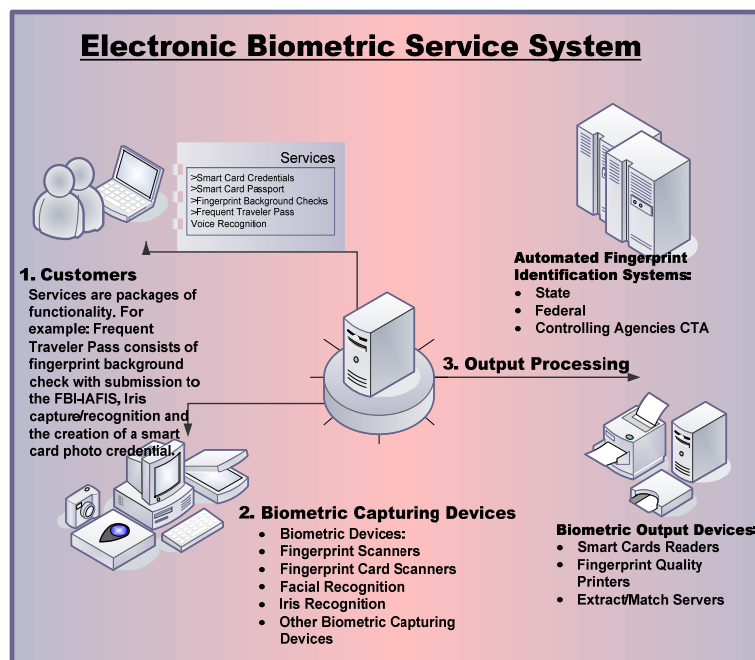


Figure 9. EBS system process flow chart.

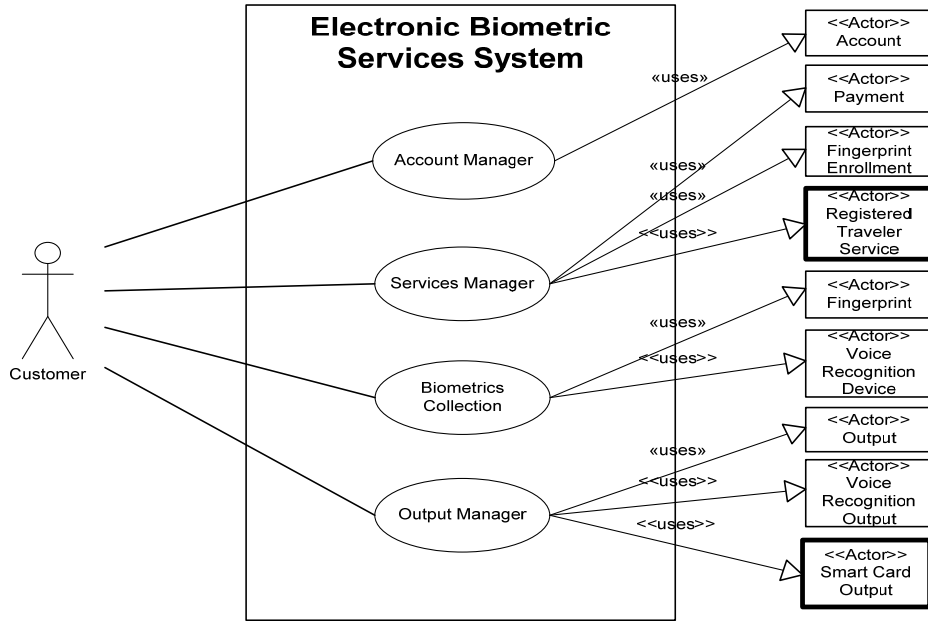


Figure 10. EBS system use case model.

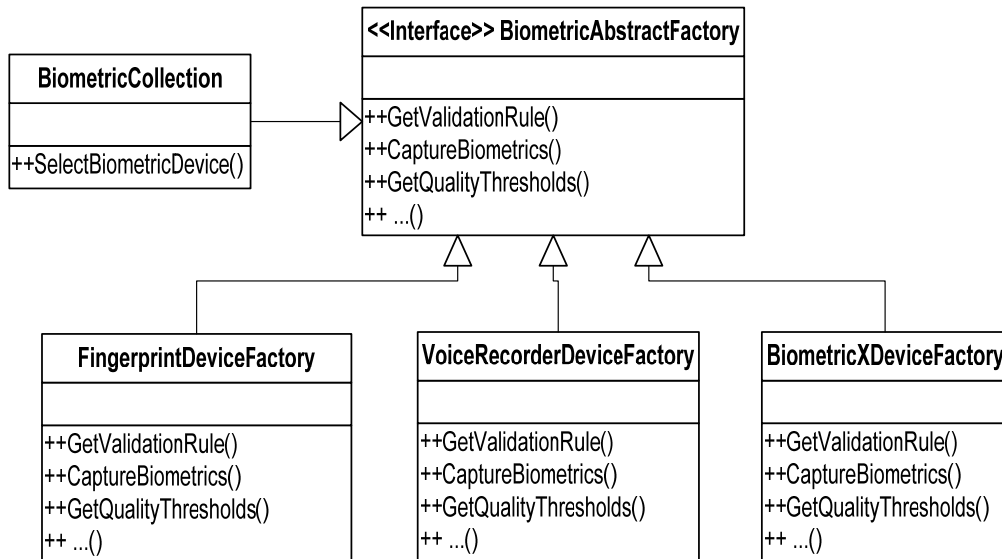


Figure 11. EBS biometric abstract factory design pattern.

Other biometrics can be easily added to these abstract classes and all biometric subclasses can be maintained independently. As also shown in Figure 11 a new biometric application X can be added to the existing system without complications to other existing classes.

The builder design pattern separates the construction of a complex object from its representation so that the same construction process can create different representations. There are various reasons for using the builder pattern. For instance, builder design pattern can be applied to reduce the size of a class that has many methods or to facilitate step-by-step constructing of a target object. The builder pattern was selected here to reconstruct the output manager class. As reflected in Figure 12, it provided flexibility for creating different output based on the device type.

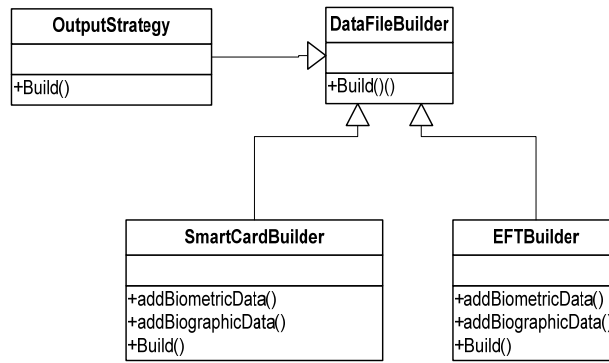


Figure 12. Biometric builder pattern for flexible output construction.

The strategy design pattern defines a family of algorithms, encapsulates each one, and makes them interchangeable. Strategy lets the algorithm vary independently from clients that use it. Strategy also separates the logic for selecting a strategy from the strategies themselves. Strategy pattern helps organize and simplify code by encapsulating different approaches to a problem in different classes. The specification classes for the creation of the different output file can be encapsulated on a strategy pattern. Figure 13 shows that those classes associated with each corresponding output. The strategy pattern makes the access to specifications for a data format more efficient.

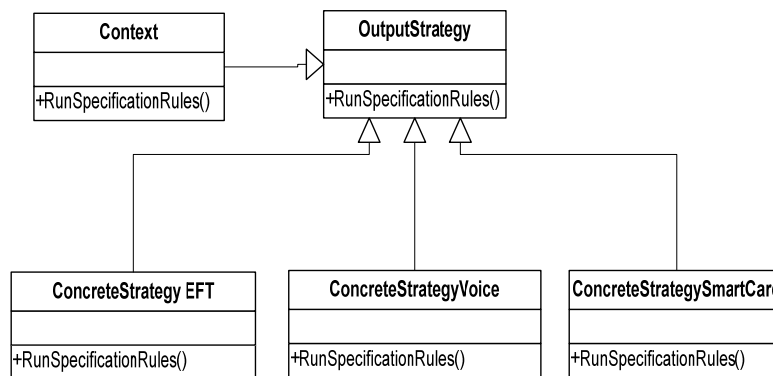


Figure 13. Output strategy pattern for specification rules handling.

### Case Study Summary

Previous two sections demonstrated how a well-designed system using object-oriented and pattern design can help achieve management flexibility, objects reusability (agility), and manageability of a complex system.

Requirement management is made easier due to object reuse and the ability to manage functionality through patterns. As demonstrated the object design for voice recognition and future development is made easier because new requirements can be handled by extending existing design patterns without change any part of the existing structure and computer codes. Design patterns have proven to provide the system model with greater flexibility and agility than what could be provided by basic object-oriented design approaches alone.

### Conclusions

In this paper the concept of object-oriented analysis and design patterns was used to analyze an EBS system model. This is a system that combines common requirements among the biometric applications. Two

models were created, one of which represents object-oriented analysis and design (EFS system). It provided some reusability of the objects and classes. The second model (EBS system) consisted additional requirements other than those analyzed in the EFS system model. This analysis and design interaction leads to the application of the design patterns by reusing the object created in the EFS model. Two case studies demonstrate how design patterns applied to the EBS model can help achieve flexibility and object reusability (agility) while maintaining the integrity and security of the main system model.

### References

- Criminal Justice Information Services Division (CJIS). (1999). Electronic fingerprint transmission specification. Department of Justice FBI.
- Davide, M. (2003). *Handbook of fingerprint recognition*. New York: Springer.
- Gomaa, H. (2004). *Designing software product lines with UML: From use cases to pattern-based software architectures*. Redwood City, California: Addison-Wesley.
- Integrated Automated Fingerprint Identification System (IAFIS). (2010). Retrieved from <http://www.fbi.gov/hp/cjisd/iafis.html>
- Komarinski, P. (2005). *Automated fingerprint identification systems (AFIS): Fingerprints-identification-data processing, LAW/forensic science*. Amsterdam: Elsevier Academic.
- Larman, C. (2002). *Applying UML and patterns: An introduction to object-oriented analysis and design and the unified process*. Upper Saddle River, N.J.: Prentice Hall.
- Maio, D. A. (2002). Smart ID in Europe: Different views, uncertain future: Strategic planning. SPA-15-4207 Research Note, Gartner Research.
- Moenssens, A. (2006). Is fingerprint identification a science: Authors' response to letter. *Journal of Forensic Identification*, 56(4), 500-510.
- NIST. (2000). American national standard for information systems—Data format for the interchange of fingerprint, Facial & Scar Mark & Tattoo (SMT) information. NIST Special Publication.
- NIST. (2001). Common Biometric Exchange File Format (CBEFF). NIST, US Department of Commerce.
- Unified Modeling Language (UML). (2009). *Object management group*. Retrieved from <http://www.omg.org/technology/documents/formal/uml.htm>
- Yacoub, S., & Ammar, H. H. (2004). *Pattern-oriented analysis and design: Composing patterns to design software systems*. Boston: Addison Wesley.