

БИОМЕТРИЧЕСКИЕ ИНТЕРФЕЙСЫ ПРИКЛАДНОГО ПРОГРАММИРОВАНИЯ

Е.В. Медведев

В статье описываются биометрические интерфейсы прикладного программирования, их достоинства и недостатки.

Введение

Для распространения практики применения биометрических технологий необходимо выработать общие стандарты интерфейса, а также методы работы и подсчётов. К сожалению, устройства для биометрических измерений все ещё остаются несовместимыми друг с другом, за исключением случаев, когда используются общепринятые стандарты. Несколько стандартов в данный момент находятся на стадии разработки и тестирования. Например, BioAPI [1] является стандартом интерфейса прикладного программирования, который позволяет отделить биометрические технологии от использующих их приложений.

1. Интерфейсы прикладного программирования

В биометрии, как в любой быстро развивающейся области технологий, требуются стандарты, которые обеспечили бы совместимость устройств, а также интерфейсы прикладного программирования (ИПП). В обычной биометрической системе распознавания существует три уровня взаимодействия.

Нижний уровень — это аппаратные средства, специфичные для биометрических систем. Также для этого уровня должны быть разработаны стандарты портов, шнуров и т. д., которые используются в том или ином случае. Следующий уровень относится к процедуре распознавания основного биометрического сигнала с шаблонами, определяемыми поставщиком. Самый высокий уровень — это уровень приложения. Стандарты должны применяться на каждом уровне [2]. В сфере стандартизации биометрических интерфейсов прикладного программирования (ИПП) существует еще много нерешенных проблем. При всех существующих биометрических протоколах, особенно динамических диалоговых протоколах, очевидно, что это очень сложная задача. Ниже приведены несколько примеров ИПП.

2. BioAPI

Есть две различные версии BioAPI: американский стандарт (ANSI/INCITS 358-2002, также известный, как BioAPI 1.1) и международный стандарт (ISO/IEC 19794-1:2005, также известный, как BioAPI 2.0).

BioAPI 1.1 идентичен спецификации, изначально разработанной BioAPI консорциумом. В отличие от него, BioAPI 2.0 — совершенно новая версия, разработанная международным комитетом по стандартизации ISO для биометрии (ISO/IEC JTC1 SC37).

Есть много различий между BioAPI 1.1 и BioAPI 2.0. Предыдущая компонентная модель из BioAPI 1.1, которая может быть описана в терминах приложения, взаимодействующего с одним или несколькими провайдерами биометрических служб (BSP) в пределах фреймворка, была сохранена и доработана. Есть несколько новых функциональностей в BioAPI 2.0, но кое-что было удалено по сравнению с BioAPI 1.1. Другими словами, BioAPI 2.0 — это результат некоторых упрощений и усовершенствований, сделанных над BioAPI 1.1.

Новый стандарт предоставляет больше возможностей для разработчика, большую совместимость и удобство в использовании, по сравнению с предыдущей версией. Пример новой функциональности в BioAPI 2.0 — это возможность для BSP делегировать часть своих обязанностей другим компонентам, называемым провайдерами биометрических функций (BFP), таким образом обеспечивая возможность разделения системы на несколько программных продуктов. Пример старой функциональности, которая была удалена, — механизм потоковых обратных вызовов, обеспечивающих взаимодействие BSP друг с другом.

3. Различия между BioAPI 1.1 и BioAPI 2.0

Компонентная модель

Основное различие между двумя версиями — это их компонентная модель. Компонентная модель BioAPI 1.0 содержит фреймворк, один или несколько BSP, используемых фреймворком и одним приложением, использующим фреймворк. Эталонная реализация BioAPI 1.1 поддерживает ровно одно приложение для одного фреймворка. Экземпляр фреймворка не может существовать без приложения, которое загружает его в оперативную память, и разные приложения используют разные экземпляры фреймворка. Если два приложения запрашивают один и тот же BSP, будут созданы разные экземпляры служб независимо в каждом из приложений.

Существует 4 категории BFP.

1. Сенсорные BFP ответственны за управление сенсорами, обычно являющимися физическими устройствами, способными записывать биометрические образцы.
2. Архивные BFP ответственны за управление доступом к базе данных биометрических образцов, которая может содержаться либо на смарт-карте,

либо в реляционной базе данных, либо в любом другом доступном источнике.

3. Процессинговые BFP ответственны за обработку биометрических образцов; преобразуют их из «промежуточных» в «обработанные».
4. Сопоставляющие BFP ответственны за сравнение биометрических образцов с извлечёнными из базы данных эталонами и возвращение результата сравнения.

Так же, как и BioAPI 1.1, BioAPI 2.0 определяет два программных интерфейса, один из которых предназначен для взаимодействия приложения с фреймворком (BioAPI API), другой — для взаимодействия фреймворка с BSP (BioSPI API). Некоторые функции уникальны в каждом из интерфейсов, но большинство функций имеет соответствующий аналог в обоих интерфейсах.

Тем не менее в BioAPI 2.0 существует дополнительный уровень интерфейсов для BFP, называемых провайдерами функций (FPI). В отличие от BioAPI API и BioSPI API они различны для каждой категории BFP. BioAPI 2.0 состоит из двух частей. Первая часть, часто называемая «BioAPI 2.0 standard», не определяет какие-либо FPI, хотя описывает компонентную модель и 4 категории BFP. Другая часть описывает FPI для остальных категорий BFP.

Ниже перечислены некоторые компоненты, поддерживаемые новой моделью BioAPI 2.0.

1. Специальный компонент (BFP), который осуществляет запись образцов и может быть использован службой, не знающей деталей реализации данного компонента.
2. Специальный компонент (BFP), реализующий алгоритм обработки и алгоритм сопоставления для определённого типа биометрических данных, который может быть использован биометрическими службами (BSP) других производителей. Таким образом разделяются обязанности между компонентами.
3. Специальный компонент, управляющий доступом к базе данных биометрических образцов, который также может быть использован биометрическими службами.

Надо заметить, что спецификация BioAPI не накладывает строгих ограничений на то, чтобы все службы были разбиты на BSP и набор BFP. Несмотря на то, что новая компонентная модель даёт возможность разделения обязанностей между различными компонентами, существуют случаи, когда подобное разделение невозможно либо приведёт к уменьшению производительности. Например, BSP, осуществляющая идентификацию, может быть жёстко связана с базой данных и алгоритмом сопоставления, и вынесение определённой функциональности в отдельные компоненты может сделать невозможным взаимодействие всех частей.

Сессии, устройства и юниты

В BioAPI 1.1 приложение должно «присоединить» BSP, предварительно его загрузив, прежде чем оно сможет использовать какие-либо функции данной службы. Соответственно, приложение должно «отсоединить» BSP, когда работа со службой завершена.

BioAPI 1.1 поддерживает управление несколькими устройствами, ассоциированными с биометрическими сенсорами, и вызов функции BioAPI_ModuleAttach должен содержать идентификатор устройства, используемого в конкретной сессии. Алгоритмы обработки и сопоставления являются встроенными компонентами BSP, и не существует возможности выбора конкретного алгоритма в приложении. Доступ к базе данных также является встроенной компонентой без возможности выбора в приложении. В BioAPI 2.0 термин устройство было замещено термином юнит, который является обобщением предыдущего. Существует 4 категории юнитов.

1. Сенсорный юнит представляет собой сенсор – физическое устройство, способное записывать биометрические данные.
2. Архивный юнит представляет собой модуль доступа к базе данных биометрических образцов.
3. Процессинговый юнит представляет собой алгоритм обработки биометрических данных.
4. Сопоставляющий юнит представляет алгоритм сравнения биометрических образцов. Существует прямое соответствие категорий юнитов и категорий BFP (см. выше).

В BioAPI 2.0 BSP может поддерживать несколько сенсоров, несколько баз данных, несколько обрабатывающих алгоритмов и несколько сопоставляющих алгоритмов. Таким образом, юниты представляют собой модель реальных объектов рассматриваемой системы в рамках BioAPI.

В BioAPI 2.0 существует понятие сессии — набора юнитов, который может содержать не более одного юнита каждой категории. Например, рассмотрим BSP, который поддерживает 2 сенсорных юнита, 2 архивных юнита, 2 процессинговых юнита и 2 сопоставляющих юнита. Приложение может создать сессию, состоящую из одного сенсорного юнита и одного процессингового юнита, и другую сессию, состоящую из одного процессингового юнита, одного сопоставляющего юнита и одного архивного юнита. Сессия не может содержать два юнита одной и той же категории, например, два сенсорных юнита.

В BioAPI 2.0 так же, как и в BioAPI 1.1, сессия необходима для приложения, чтобы оно могло делать вызовы определённых функций, например, "BioAPI_VerifyMatch". Но сессия теперь не является состоянием BSP, как это было в BioAPI 1.1, а является набором ресурсов (т.е. набором юнитов), которые будут использованы в биометрических операциях. В BioAPI 2.0 функция "BioAPI_BSPAttach" должна вызываться со списком идентификаторов юнитов в

качестве параметров. Каждый из идентификаторов либо явным образом определяет конкретный юнит нужной категории, либо неявным образом определяет юнит по умолчанию, который BSP должна использовать, либо указывает, что ни один из юнитов данной категории не должен быть использован в сессии.

Реестр, установка компонентов, перечень доступных компонентов

В BioAPI 1.1 существует реестр, содержащий информацию о фреймворке, информацию о доступных службах (BSP) и информацию о доступных устройствах, управляемых этими службами. Однако, есть только одна функция, обращающаяся к реестру: `BioAPI_EnumModules`. Эта функция возвращает информацию о доступных службах. Не существует функции, которая бы возвращала информацию о доступных устройствах. Кроме того, хотя BioAPI требует, чтобы установщики служб передавали информацию о службах в реестр, не существует стандартной функции или механизма, который обеспечивал бы подобную передачу. Отсутствие подобного механизма в стандарте является проблемой, так как это вынуждает разработчиков фреймворка реализовывать реестр таким же образом, как он реализован в эталонной версии, что добавляет ненужную сложность. Кроме того, разработчики должны либо связывать их код с бинарными библиотеками, являющимися частью эталонной реализации, либо даже эмулировать их работу так, чтобы была возможность добавлять информацию о службах в реестр, как того требует стандарт.

В BioAPI 2.0 эта проблема была решена простым способом. Реестр теперь полностью инкапсулирован в фреймворк и доступен только посредством небольшого набора стандартных функций. BioAPI API включает функции для установки и удаления BSP и BFP: `BioAPI_Util_InstallBSP` и `BioAPI_Util_InstallBFP`, стандарт также включает полный набор функций, возвращающих информацию о доступных службах (`BioAPI_EnumBSPs`), доступных функциях (`BioAPIEnumBFPs`) и о самом фреймворке (`BioAPI_GetFrameworkInfo`).

Единственный, кто должен понимать внутреннюю структуру реестра или использовать двоичные библиотеки, — это разработчики фреймворка. Разработчики приложений, служб (BSP) и функций (BFP) не должны обращаться к реестру другими способами, кроме как посредством стандартных функций.

Расширения и дополнения к BioAPI

Существуют стандарты, являющиеся дополнительным слоем к BioAPI и расширяющие его возможности.

Например, биометрический протокол взаимодействия (BIP, standard ISO/IEC 24708) расширяет BioAPI 2.0 и позволяет приложению, запущенному на одной машине, использовать службы (BSP), запущенные на других машинах так, как будто они запущены локально. Другой пример — BioGUI, дополнение к BioAPI 2.0, обеспечивающее графический интерфейс к приложению и позволяющее контролировать поток данных.

4. Windows Biometric Framework API

Windows Biometric Framework (WBF) [3] — это платформа, обеспечивающая поддержку различных биометрических технологий и устройств в операционной системе Windows и предоставляющая разработчикам приложений набор ИПП для доступа к биометрическим устройствам, осуществления идентификации и аутентификации пользователей (операционная система Windows 7 поддерживает только сканер отпечатка пальца в качестве биометрического устройства). WBF состоит из четырёх компонентов:

1. Базовая платформа, включающая описание интерфейса драйверов, платформу для подключаемых расширений, клиентское API.
2. Компоненты взаимодействия с пользователем, обеспечивающие унифицированный интерфейс с пользователем в ОС Windows.
3. Компоненты для управления и конфигурирования биометрических устройств.
4. Компоненты для распространения биометрических драйверов и компонентов.

Базовая платформа WBF

Базовая платформа WBF состоит из следующих компонентов:

- 1) интерфейс драйверов биометрических устройств (WBDI);
- 2) биометрическая служба (WBS);
- 3) интерфейс прикладного программирования (WBF API).

WBDI определяет общий интерфейс, который должны реализовывать драйверы биометрических устройств для осуществления взаимодействия биометрической службы с устройством. Биометрическая служба обеспечивает доступ только к тем устройствам, которые реализуют указанный интерфейс. Это позволяет приложениям использовать биометрические устройства через WBF API. Биометрическая служба (WBS) — это компонент, который управляет биометрическими устройствами через WBDI драйвер и предоставляет к ним доступ для приложений с помощью WBF API.

WBS обеспечивает конфиденциальность данных пользователя, разделяя клиентское приложение и биометрические данные. А именно, WBS выступает в роли посредника между приложением и биометрическим устройством и осуществляет операции записи, обработки и сохранения биометрических данных.

Так же, как и в BioAPI 2.0, существует механизм разделения функциональности на компоненты. В частности, WBS выполняет свою работу с помощью провайдеров биометрических служб (BSP) — аналог BFP в BioAPI 2.0. Другой особенностью WBS является унификация поведения биометрических устройств, осуществляемая посредством создания абстрактной единицы

для каждого устройства, называемой биометрическим юнитом. Биометрический юнит может осуществлять операции записи, обработки и сохранения биометрических данных. Биометрический юнит аналогичен юнитам BioAPI 2.0. Биометрический юнит состоит из трёх присоединяемых компонентов, называемых адаптерами.

1. Сенсорный адаптер осуществляет запись биометрических образцов.
2. Базовый адаптер осуществляет операции нормализации, извлечения параметров, генерации биометрических эталонов, сопоставления образцов с эталонами.
3. Архивный адаптер осуществляет операции сохранения и извлечения биометрических образцов.

Таким образом, набор адаптеров играет роль сессии BioAPI 2.0.

WBS предоставляет доступ к биометрическим устройствам посредством WBF API. Интерфейс позволяет приложениями регистрировать, идентифицировать и верифицировать пользователей. Кроме того, WBF API позволяет:

- 1) запрашивать информацию о биометрических устройствах;
- 2) осуществлять поиск устройства в системе;
- 3) управлять сессиями пользователей;
- 4) контролировать события в WBF;
- 5) сохранять биометрические образцы.

Сравнивая BioAPI и Windows Biometric Framework, можно сказать, что они имеют схожую архитектуру и компонентную модель. Оба интерфейса предоставляют возможности для разбиения приложения на компоненты (BSP и BFP в BioAPI, BSP в Windows Biometric Framework) и высокий уровень абстракции, что делает приложение легко расширяемым и переносимым. Использование этих интерфейсов делает приложения, использующие биометрические технологии, более совместимыми друг с другом.

ЛИТЕРАТУРА

1. BioAPI Consortium. BioAPI Specification Version 1.1, 2001. URL: <http://www.bioapi.org/Downloads/BioAPI~1.1.doc> (дата обращения: 10.10.2011)
2. Болл Руд М. и др. Руководство по биометрии. М. : Техносфера, 2007. 368 с.
3. Microsoft corporation. Introduction to the Windows Biometric Framework (WBF), 2010. URL: <http://msdn.microsoft.com/en-us/windows/hardware/gg463089> (дата обращения: 10.10.2011)