BIOMETRIC SYSTEM BY POINTING FINGER

Eugène MBUYI MUKENDI[1], Jean Didier BATUBENGA MWAMBA[2]

[1] Professor University of Kinshasa, Computer Science Department, DR Congo

[2] University of Kinshasa, Computer Science Department, DR Congo

Kinshasa Computer Science Laboratory.

Abstract

Biometrics is an important application of pattern recognition, the cross-domain path between data analysis and artificial intelligence. Technological change and the growing computerization of society have paved the way for biometric systems. They rest on sequence procedures that always follow the same development: biometric data is converted to digital data and stored in memory serve as biometric reference of the individual concerned. A biometric system helps to recognize a person whose identity has been previously stored in a database. In this paper, we present an automatic pointing system which allows identifying personnel from his fingerprint to know the time he/she attaches to the work. It is an authentication work aiming at verifying the identity of personnel during pointing in a company.
Fingerprint recognition's algorithm has been suggested by A.K. Jain who is probably the most known.

Keywords: biometrics, pattern recognition, identification, authentication.

INTRODUCTION [ABD1], [BAT2], [BOU3], [CHA4]

I.1. Biometrics

Biometrics is a branch of pattern recognition which always generates more interest. Biometric systems help to automatically identify people using physical or behavioral characteristics such as face, fingerprint, signature or walking. In other words, biometric refers to the identification of an individual based on morphological, biological or behavioral characteristics. These biometric characteristics are unique and specific to an individual and there is little possibility that other people may replace them. It is then more powerful in terms of safety.
Biometric data analysis can be classified into three broad categories:
• Analysis based on morphology: fingerprint, hand geometry, facial features, vein of the retina, iris, etc…
• Analysis of biological evidence: saliva, smell, urine, blood, DNA, etc...
• Analysis based on behavior: signature tracing dynamic, hits on a computer keyboard, voice, etc ...
These biometric features cannot be easily stolen, falsified or shared.

They must be reliable and secure for the recognition of people than traditional methods. Therefore, they must meet certain criteria for a broad reliability of biometric systems. They must be:
• Universal: must exist in every person
• Unique: to help distinguish one individual from another
• Permanent: stability over time
• Measurable: sensor technology exists that allows a comparison to the days to come
• Recordable
• Unfalsifiable.

I.2. Biometric system

I.2.1. Operating modes of a biometric system

Overall, a biometric system consists of two phases: an enrollment phase and a recognition phase.
Enrollment

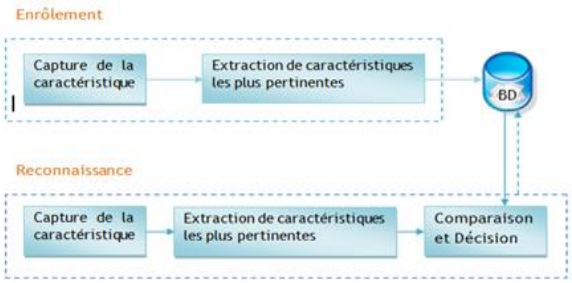

Figure 1. Architecture a biometric system

- ✓ **Enrollment**

This phase consists of creating a biometric template of an individual who will serve as a reference for recognition. For this purpose, the biometric characteristic of the individual is captured

by a biometric sensor, and then represented in digital form and then stored in a database. Ordinarily, we do not work directly on these raw data because they contain unnecessary information to recognition. Indeed, from this data, we do extract just the “relevant" parameters, this helps to significantly reduce the size of backup data.

- ✓ **Recognition**

During the recognition phase, the biometric characteristic is captured and the relevant parameters are extracted as in the enrollment phase. The follow-up of the recognition depends on the operating mode of the system. If you are in identification mode, the system will compare the captured signal with all templates contained in the database. Then it will pull the nearest signal template to answer question such as: *"Who am I?"* It is a very difficult task because the database may contain thousands of individual information. You lose a lot of time calculating all the possible comparisons. On the other hand, in verification mode (authentication), the system will compare the signal with one of the templates in the database with the aim of answering the question: "Am I the person I pretend to be?"

I.2.2. Evaluation of performance of a biometric system

Unfortunately, biometrics has a major drawback. In fact, none of the measures used reveals to be totally accurate because it is indeed a major feature of any living organism: it adapts to the environment, grows old, undergoes more or less important trauma, and therefore the measurements change. But it must nevertheless do well to gain recognition, and in reality it will work in most cases because the system allows a margin of error between the measurement and reference. Indeed, manufacturers, developers do not seek absolute security, they want something that works in practice. Thus we find the definition of the reliability of a biometric recognition system which is characterized by two measures of error. These two measures are the False Rejection Rate (FRR) and False Acceptance Rate (FAR)). Manufacturers, developers of biometric recognition systems are seeking to reduce the FRR, while maintaining a relatively low rate of FAR. Therefore a compromise needs to found between FRR and FAR. In fact, these two measures of error are related, and there exists an equilibrium point between both of them for each biometric system. The graph, in which the two curves called ROC (Receiver Operating Characteristic) are shown, is purely demonstrative. Indeed, each measurement is based on the decision threshold of the system, a value determined by the designer or the system operator, who defines point when a match is made. Delta (Δ) represents the margin of error allowed by the system, varying from 0 to infinity. The more the allowable error margin is important, the more the false acceptance rate increases, that is to say we will accept more and more people who are not authorized (and therefore the safety of the system decreases).

However, if the rejection rate of authorized persons also decreases, this makes the system more functional and better meets user expectations. At the other extremity, if the error margin accepted by the process of biometric measurement is decreased the rate of the two trends are reversed: we will less and less accept people trying to cheat but we ill also, at the same occasion, have a rejection rate on authorized persons which will be too large to be tolerated in most cases. The usual compromise is to consider the junction of the curves, that is to say the x point where FAR and FRR are equal and at the same time to a minimum. The more x value is low, the more the system is reliable, because there is a good equilibrium of sensitivity. Besides these two error rate, FAR and FRR, the failure rate to enrollment is equally used to determine the accuracy of a biometric system.

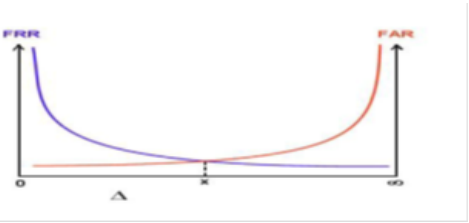

Figure 2. ROC Curve (Receiver Operating Characteristic)

- The Rate of False Rejection (FRR): it expresses the percentage of legitimate users rejected.
- The False Acceptance Rate (FAR): it expresses the percentage of impostors accepted.

$$\boldsymbol{FFR} = \frac{false\ rejections\ number}{clients\ number}$$

$$\boldsymbol{FAR} = \frac{false\ accepatations\ number}{impostors\ number}$$

I.2.3. Technique based on fingerprint

This is the oldest and most widely used identification technology. It represents more than forty-five percent of applications. The basic data in the case of the fingerprint is design represented by the ridges and valleys of the skin. This design is unique and different for each individual. In practice, the extraction of biometric data using this technique involves the imposition of a finger on a fingerprint reader.

Figure 3 biometric fingerprint reader

I.3. Fingerprints Recognition

The first traces of use of fingerprints were discovered in Egypt and date back to the time of the pyramids, more than 4,000 years ago. The Chinese have also used this medium very early to sign official documents, but they probably do not know that fingerprints were unique to each person and allow reliable identification. In 1856 the Englishman **William Hershel**, after using the fingerprint as a signature on Indian population he headed, began to realize that fingerprints were unique and constant over time.

In 1888, the British **Francis Galton** published a study on fingerprints where he established their characteristics (the notions of uniqueness, invariance, minutiae, classification). In 1901 the technique of identification through fingerprints was officially adopted in England in the judicial system. This technique was then greatly expanded in criminal investigations.

II. IMPLEMENTATION OF THE SYSTEM [CHA5], [CHA6], [MBU10]

❖ **Identification**

Each person must be characterized by biographical data (last name, middle name, first name, date of birth, sex, address, nationality, etc...) and biometric data (fingerprint).

❖ **Fingerprint Capture (Enrollment)**

The biometric data is captured directly from a fingerprint reader. The system should only allow saving the information necessary for the recognition and not to make the data base heavy.

❖ **Search**

Various registered persons can be identified from the data managed by the system. No matter what data it is biographical or biometric it can provide research services. We then speak of matching when it comes to biometric data.

❖ **Payment Processing**

The system must be able to calculate wages and provide employees payroll.

❖ **Sending SMS**

The system must be capable of processing payment. By the moment all calculations were well made, the system must be capable from a click to send SMS to the relevant staff to inform them of their situation.

❖ **Statistics' Treatment**

The system must be able to produce statistics on all employees. The number of days when one actually worked per month, per year, etc...

II.1. Collection of operational needs

❖ **Security**

It is the system administrator which manages the access rights. When connected, each user must be recognized from the system and by his/her fingerprint and access only information that is necessary for him depending on the rights that they have.

II.2. Description of the system context

Now that the collection of needs is carried out, description of the context of the system can begin. It consists of three successive operations:

- Identification of actor,
- Messages identification,
- The realization of context diagrams.

II.3 Identification of actors

An **actor** is a user who always has the same behavior vis-à-vis a use case. This is an external component (hardware or other system device) which interacts with the system or a role that a user plays in relation to the system. A single user can play multiple roles as multiple users can play the same role. For example, the administrator of a messaging system can also be a user of the same messaging. He will be considered, as an actor of the system, in the administrator role on the one hand and that of user on the other hand.

Considering this definition, we identify the following actors:

Actors	Role
Administrator	Manages access to all configuration settings of the application. He handles the recruitment of employees. It processes the payment and prepares payroll. He consults statistics on all staff.
User	Role played by all those who have access to the application, but not accessing all the features of the application.
Fingerprint reader	It captures digital fingerprints
GSM Modem	Send SMS to personnel

Table 1: Identification of actors

II.4 Functional needs Capture

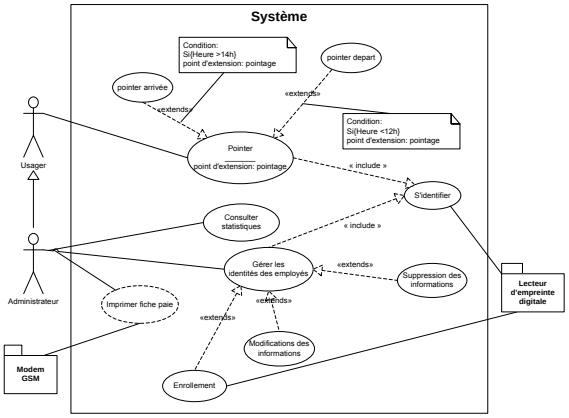

Figure.4 diagram of use cases

II.5. Application Presentation

To avoid overloading the paper, we present a number of screen captures of our application.

A simple click on the executable of the application launches a splash screen. For security reasons when the application is provided for the first time, it contains no information. Thus we have provided a form of authentication where the user enters the password and login.

Figure 5. Application Launch Screen

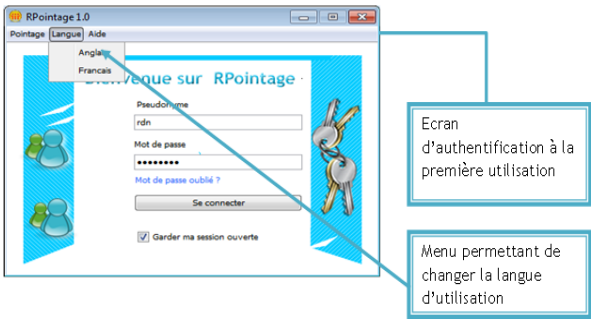

Figure 6. Authentication Screen

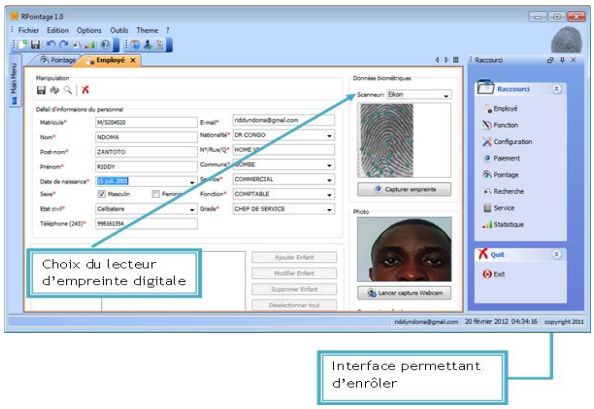

Figure 7. Enrollment Form

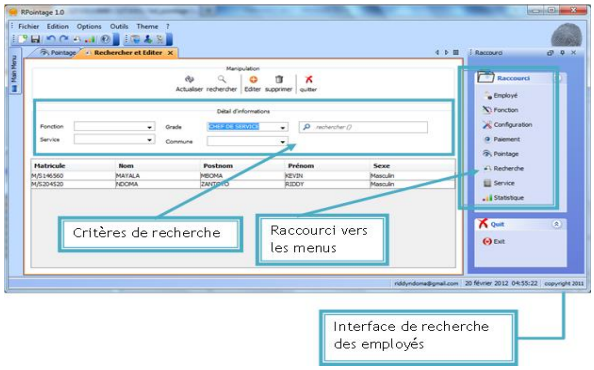

Figure 8. Search Form

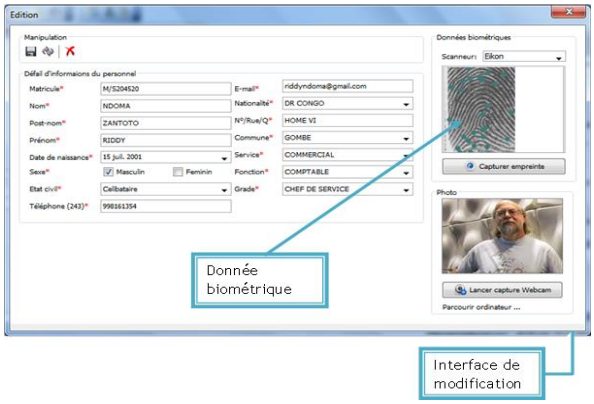

Figure 9. Change Form

II.6 SOURCES CODES EXTRACT

A: Database Scripts

```
-- Base de donnÃ©es: `bd_pointage`
-- ------------------------------------------
--------------
-- Structure de la table `t_commune`
CREATE TABLE IF NOT EXISTS `t_commune` (
`id_commune` int(11) NOT NULL AUTO_INCREMENT,
`nom_commune` text NOT NULL,
PRIMARY KEY (`id_commune`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
AUTO_INCREMENT=131341 ;
-- ------------------------------------------
--------------
-- Structure de la table `t_employe`
CREATE TABLE IF NOT EXISTS `t_employe` (
`matricule` varchar(11) NOT NULL,
`nom_employe` text NOT NULL,
`postnom_employe` text NOT NULL,
`prenom_employe` text NOT NULL,
`date_naissance` date NOT NULL,
`sexe_employe` text NOT NULL,
`etat_civil` text NOT NULL,
`nationalite` text NOT NULL,
`phone_employe` int(11) NOT NULL,
`mail_employe` text NOT NULL,
`rue_employe` text NOT NULL,
`photo_employe` longblob NOT NULL,
`empreinte_mploye` longblob NOT NULL,
`id_commune` int(11) NOT NULL,
`id_service` varchar(11) NOT NULL,
`id_fonction` int(11) NOT NULL,
`id_grade` int(11) NOT NULL,
PRIMARY KEY (`matricule`),
KEY `id_commune`
(`id_commune`,`id_service`,`id_fonction`,`id_
grade`),
KEY `id_commune_2`
(`id_commune`,`id_service`,`id_fonction`,`id_
grade`),
KEY `id_service` (`id_service`),
KEY `id_fonction` (`id_fonction`),
KEY `id_grade` (`id_grade`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-- ------------------------------------------
--------------
-- Structure de la table `t_enfant`
CREATE TABLE IF NOT EXISTS `t_enfant` (
`id_enfant` int(11) NOT NULL AUTO_INCREMENT,
`prenom_enfant` text NOT NULL,
`nom_enfant` text NOT NULL,
`date_naissance_enfant` date NOT NULL,
`matricule` varchar(11) NOT NULL,
PRIMARY KEY (`id_enfant`),
KEY `matricule` (`matricule`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
AUTO_INCREMENT=5 ;
-- ------------------------------------------
--------------
-- Structure de la table `t_fonction`
CREATE TABLE IF NOT EXISTS `t_fonction` (
`id_fonction` int(11) NOT NULL,
`nom_fonction` text NOT NULL,
`salaire_base` double NOT NULL,
`transport` double NOT NULL,
`logement` double NOT NULL,
`autres` double NOT NULL,
PRIMARY KEY (`id_fonction`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-- ------------------------------------------
--------------
-- Structure de la table `t_grade`
CREATE TABLE IF NOT EXISTS `t_grade` (
`id_grade` int(11) NOT NULL AUTO_INCREMENT,
`nom_grade` text NOT NULL,
PRIMARY KEY (`id_grade`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
AUTO_INCREMENT=131370 ;
-- ------------------------------------------
--------------
-- Structure de la table `t_jourpointage`
CREATE TABLE IF NOT EXISTS `t_jourpointage` (
`id_jourpointage` int(11) NOT NULL
AUTO_INCREMENT,
`date_jourpointage` date NOT NULL,
PRIMARY KEY (`id_jourpointage`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
AUTO_INCREMENT=11 ;
-- ------------------------------------------
--------------
-- Structure de la table
`t_participerpointage`
CREATE TABLE IF NOT EXISTS
`t_participerpointage` (
`matricule` varchar(11) NOT NULL,
`id_pointage` int(11) NOT NULL,
`heure_arrive` date NOT NULL,
`heure_depart` date NOT NULL,
PRIMARY KEY (`matricule`,`id_pointage`),
KEY `id_pointage` (`id_pointage`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-- ------------------------------------------
--------------
-- Structure de la table `t_service`
CREATE TABLE IF NOT EXISTS `t_service` (
`id_service` varchar(11) NOT NULL,
`nom_service` text NOT NULL,
PRIMARY KEY (`id_service`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-- ------------------------------------------
--------------
-- Structure de la table `t_user`
CREATE TABLE IF NOT EXISTS `t_user` (
`id_user` int(11) NOT NULL AUTO_INCREMENT,
`prenom` text NOT NULL,
`nom` text NOT NULL,
`postnom` text NOT NULL,
`email` text NOT NULL,
`nom_utilisateur` text NOT NULL,
`mot_passe` text NOT NULL,
`question` text NOT NULL,
`reponse` text NOT NULL,
`type` varchar(20) NOT NULL,
`etat` varchar(20) NOT NULL,
PRIMARY KEY (`id_user`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
COMMENT='table contenant les utilisateurs'
AUTO_INCREMENT=4 ;
-- Contraintes pour les tables exportÃ©es
-- Contraintes pour la table `t_employe`
ALTER TABLE `t_employe`
ADD CONSTRAINT `t_employe_ibfk_1` FOREIGN KEY
(`id_commune`) REFERENCES `t_commune`
(`id_commune`) ON DELETE CASCADE ON UPDATE
CASCADE,
ADD CONSTRAINT `t_employe_ibfk_2` FOREIGN KEY
(`id_service`) REFERENCES `t_service`
(`id_service`) ON DELETE CASCADE ON UPDATE
CASCADE,
ADD CONSTRAINT `t_employe_ibfk_3` FOREIGN KEY
(`id_fonction`) REFERENCES `t_fonction`
(`id_fonction`) ON DELETE CASCADE ON UPDATE
CASCADE,
ADD CONSTRAINT `t_employe_ibfk_4` FOREIGN KEY
(`id_grade`) REFERENCES `t_grade`
(`id_grade`) ON DELETE CASCADE ON UPDATE
CASCADE;
-- Contraintes pour la table `t_enfant`
ALTER TABLE `t_enfant`
ADD CONSTRAINT `t_enfant_ibfk_1` FOREIGN KEY
(`matricule`) REFERENCES `t_employe`
```

```
(`matricule`) ON DELETE CASCADE ON UPDATE
CASCADE;
-- Contraintes pour la table
`t_participerpointage`
--TER TABLE `t_participerpointage`
ADD CONSTRAINT `t_participerpointage_ibfk_2`
FOREIGN KEY (`id_pointage`) REFERENCES
`t_jourpointage` (`id_jourpointage`) ON
DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `t_participerpointage_ibfk_1`
FOREIGN KEY (`matricule`) REFERENCES
`t_employe` (`matricule`) ON DELETE CASCADE
ON UPDATE CASCADE;
```

B : Application Scripts

To achieve our application, we had to write more than 2000 scripts. We do put here just a part for illustrative purposes.

```
packagepointage.docking;
importcom.jidesoft.action.DefaultDockableBarD
ockableHolder;
importcom.jidesoft.docking.DockContext;
importcom.jidesoft.docking.DockableFrame;
importcom.jidesoft.docking.DockingManager;
importcom.jidesoft.document.DocumentPane.Tabb
edPaneCustomizer;
importcom.jidesoft.utils.PortingUtils;

public class MainFrame extends
DefaultDockableBarDockableHolder {
publicMainFrame(String title) throws
HeadlessException {
super(title);
newConnexion();
}
publicMainFrame() throws HeadlessException {
this("");
}
public static
DefaultDockableBarDockableHoldershowFrame(fin
al boolean exit) throws PropertyVetoException
{
_frame = new MainFrame("RPointage 1.0");
_frame.setDefaultCloseOperation(JFrame.DISPOS
E_ON_CLOSE);
_frame.setIconImage(new
ImageIcon(_frame.getClass().getResource("/poi
ntage/ressources/note.png")).getImage());
cbf = new MainFrameCommandBarFactory(_frame);
// add a window listener to do clear up when
windows closing.
_windowListener = new WindowAdapter() {
@Override
public void windowClosing(WindowEvent e) {
super.windowClosing(e);
actionExit();
}
};
_frame.addWindowListener(_windowListener);
// set the profile key
_frame.getLayoutPersistence().setProfileKey(P
ROFILE_NAME);
_frame.getLayoutPersistence().setXmlFormat(tr
ue);
// create tabbed-document interface and add
it to workspace area
_documentPane = createDocumentTabs();
_documentPane.setTabbedPaneCustomizer(new
TabbedPaneCustomizer() {
@Override
public void customize(final
JideTabbedPanetabbedPane) {
tabbedPane.setShowCloseButtonOnTab(true);
tabbedPane.setShowCloseButtonOnSelectedTab(tr
ue);
}
});
_frame.getDockingManager().getWorkspace().set
Layout(new BorderLayout());
_frame.getDockingManager().getWorkspace().add
(_documentPane, BorderLayout.CENTER);
_frame.getDockableBarManager().setProfileKey(
PROFILE_NAME);
// add toolbar
_frame.getDockableBarManager().addDockableBar
(MainFrameCommandBarFactory.createMenuCommand
Bar());
_frame.getDockableBarManager().addDockableBar
(MainFrameCommandBarFactory.createStandardCom
mandBar());
//
_frame.getDockableBarManager().addDockableBar
(MainFrameCommandBarFactory.createOptionsComm
andBar());
_frame.getDockableBarManager().addDockableBar
(MainFrameCommandBarFactory.createToolsComman
dBar());
// add status bar
_statusBar = createStatusBar();
_frame.getContentPane().add(_statusBar,
BorderLayout.AFTER_LAST_LINE);
_frame.getDockingManager().getWorkspace().set
AdjustOpacityOnFly(true);
_frame.getDockingManager().setUndoLimit(10);
_frame.getDockingManager().beginLoadLayoutDat
a();
// add all dockable frames
_frame.getDockingManager().addFrame(MainFrame
CommandBarFactory.createFramePrincipal(_frame
));
_frame.getDockingManager().addFrame(MainFrame
CommandBarFactory.createFrameRaccourci());
_frame.getDockingManager().setShowGripper(tru
e);
_frame.getDockingManager().setOutlineMode(Doc
kingManager.TRANSPARENT_OUTLINE_MODE);
_frame.getDockingManager().setPopupMenuCustom
izer(new
com.jidesoft.docking.PopupMenuCustomizer() {
@Override
public void customizePopupMenu(JPopupMenu
menu, final DockingManagerdockingManager,
final DockableFramedockableFrame,
booleanonTab) {
menu.addSeparator();
menu.add(new AbstractAction("Move to Document
Area") {
@Override
public void actionPerformed(ActionEvent e) {
dockingManager.removeFrame(dockableFrame.getK
ey(), true);
DocumentComponentdocumentComponent = new
DocumentComponent((JComponent)
dockableFrame.getContentPane(),
dockableFrame.getKey(),
dockableFrame.getTitle(),
dockableFrame.getFrameIcon());
_documentPane.openDocument(documentComponent)
;
_documentPane.setActiveDocument(documentCompo
nent.getName());
}
});
}
});
// load layout information from previous
session
_frame.getLayoutPersistence().loadLayoutData(
);
if (Lm.DEMO) {
Lm.z();
}
```

```
_frame.toFront();
return _frame;
}
public static void actionResearch() {
if
(!_documentPane.isDocumentOpened("Rechercher
et Editer")) {
try {
_documentPane.openDocument(new
DocumentComponent(new SearchEdit(_frame),
"Rechercher et Editer", "Rechercher et
Editer"
+ "", new
ImageIcon(_frame.getClass().getResource("/ima
ges/gnome-orca12.png"))));
} catch (Exception ex) {
ex.printStackTrace();
}
}
_documentPane.setActiveDocument("Rechercher
et Editer");
}
private static void clearUp() {
_frame.removeWindowListener(_windowListener);
_windowListener = null;
if (_frame.getLayoutPersistence() != null) {
_frame.getLayoutPersistence().saveLayoutData(
);
}
if (_documentPane != null) {
_documentPane.dispose();
_documentPane = null;
}
if (_statusBar != null &&
_statusBar.getParent() != null) {
_statusBar.getParent().remove(_statusBar);
}
_statusBar = null;
_frame.dispose();
_frame = null;
}
public static StatusBarcreateStatusBar() {
// setup status bar
StatusBarstatusBar = new StatusBar();
progress = new ProgressStatusBarItem();
statusBar.add(progress, JideBoxLayout.VARY);
finalLabelStatusBarItem label = new
LabelStatusBarItem("");
label.setText("riddyndoma@gmail.com");
label.setAlignment(JLabel.CENTER);
statusBar.add(label, JideBoxLayout.FLEXIBLE);
finalTimeStatusBarItem time = new
TimeStatusBarItem();
time.setTextFormat(new SimpleDateFormat("dd
MMMM yyyyHH:mm:ss"));
statusBar.add(time, JideBoxLayout.FLEXIBLE);
//       elements added on Statusbar by RDN
finalLabelStatusBarItemmyJL = new
LabelStatusBarItem("");
myJL.setText("copyright 2011");
myJL.setAlignment(JLabel.CENTER);
statusBar.add(myJL, JideBoxLayout.FLEXIBLE);
//      Font added on Statusbar by RDN
Font myFont = new Font("tahoma", Font.PLAIN,
10);
statusBar.setFont(myFont);
returnstatusBar;
}
private static
DocumentPanecreateDocumentTabs() {
DocumentPanedocumentPane = new DocumentPane()
{
// add function to maximize (autohideAll) the
document pane when mouse double clicks on the
tabs of DocumentPane.
@Override
protectedIDocumentGroupcreateDocumentGroup()
{
IDocumentGroup group =
super.createDocumentGroup();
if (group instanceofJideTabbedPane) {
((JideTabbedPane) group).addMouseListener(new
MouseAdapter() {
@Override
public void mouseClicked(MouseEvent e) {
if (SwingUtilities.isLeftMouseButton(e)
&&e.getClickCount() == 2) {
if (!_autohideAll) {
_fullScreenLayout =
_frame.getDockingManager().getLayoutRawData()
;
_frame.getDockingManager().autohideAll();
_autohideAll = true;
} else {
if (_fullScreenLayout != null) {
_frame.getDockingManager().setLayoutRawData(_
fullScreenLayout);
}
_autohideAll = false;
}
Component lastFocusedComponent =
_documentPane.getActiveDocument().getLastFocu
sedComponent();
if (lastFocusedComponent != null) {
lastFocusedComponent.requestFocusInWindow();
}
}
}
});
}
return group;
}
};
documentPane.setTabPlacement(JTabbedPane.TOP)
;
documentPane.setPopupMenuCustomizer(new
PopupMenuCustomizer() {
@Override
public void customizePopupMenu(JPopupMenu
menu, final IDocumentPane pane, final String
dragComponentName,
finalIDocumentGroupdropGroup, booleanonTab) {
if
(!pane.isDocumentFloating(dragComponentName))
{
menu.addSeparator();
menu.add(new AbstractAction("Dock to the
Side") {
@Override
public void actionPerformed(ActionEvent e) {
DocumentComponentdocumentComponent =
pane.getDocument(dragComponentName);
if (documentComponent != null) {
pane.closeDocument(dragComponentName);
// check if the document is really closed.
There are cases a document is not closable or
veto closing happens which can keep the
document open after closeDocument call.
if
(!pane.isDocumentOpened(dragComponentName)) {
finalDockableFrame frame = new
DockableFrame(documentComponent.getName(),
documentComponent.getIcon());
frame.setTabTitle(documentComponent.getTitle(
));
frame.getContentPane().add(documentComponent.
getComponent());
frame.setInitIndex(0);
frame.setInitSide(DockContext.DOCK_SIDE_EAST)
;
frame.setInitMode(DockContext.STATE_FRAMEDOCK
ED);
_frame.getDockingManager().addFrame(frame);
```

```
_frame.getDockingManager().activateFrame(fram
e.getKey());
}
}
}
});
}
}
});
returndocumentPane;
}
private static
JComponentcreateMultiViewDocument(String
fileName) {
JideTabbedPane pane = new
JideTabbedPane(JideTabbedPane.BOTTOM);
pane.setTabShape(JideTabbedPane.SHAPE_BOX);
pane.addTab("Design",
createTextArea(fileName));
pane.addTab("HTML",
createTextArea(fileName));
return pane;
}
private static
JComponentcreateTextArea(String fileName) {
JTextArea area = new JTextArea();
Document doc = new PlainDocument();
try {
InputStream in =
MainFrame.class.getResourceAsStream(fileName)
;
if (in == null) {
in = new FileInputStream(fileName);
}
byte[] buff = new byte[4096];
intnch;
while ((nch = in.read(buff, 0, buff.length))
!= -1) {
doc.insertString(doc.getLength(), new
String(buff, 0, nch), null);
}
area.setDocument(doc);
} catch (IOException | BadLocationException
e) {
System.out.println(e.getLocalizedMessage());
}
return area;
}
@Override
protectedContentContainercreateContentContain
er() {
return new LogoContentContainer();
}
private class LogoContentContainer extends
ContentContainer {
@Override
protected void paintComponent(Graphics g) {
super.paintComponent(g);
ImageIconimageIcon = new
ImageIcon(_frame.getClass().getResource("/poi
ntage/ressources/empreinte.png"));
imageIcon.paintIcon(this, g, getWidth() -
imageIcon.getIconWidth() - 2, 2);
}
}
public static void actionNewChild() {
if
(!_documentPane.isDocumentOpened("Enfant")) {
_documentPane.openDocument(new
DocumentComponent(new NewChild(_frame),
"Enfant", "Enfant"+ "", new
ImageIcon(_frame.getClass().getResource("/poi
ntage/ressources/mini_child.png"))));
}
_documentPane.setActiveDocument("Enfant");
}
public staticvoidactionNewPartener() {
if
(!_documentPane.isDocumentOpened("Conjoint"))
{
_documentPane.openDocument(new
DocumentComponent(new NewPartener(_frame),
"Conjoint", "Conjoint"+ "", new
ImageIcon(_frame.getClass().getResource("/poi
ntage/ressources/mini_partener.png"))));
}
_documentPane.setActiveDocument("Conjoint");
}
public staticvoidactionNewWorker() {
if
(!_documentPane.isDocumentOpened("Employé"))
{
try {
_documentPane.openDocument(new
DocumentComponent(new NewWorker(_frame),
"Employé", "Employé"+ "", new
ImageIcon(_frame.getClass().getResource("/poi
ntage/ressources/mini_worker.png"))));
} catch (SQLException ex) {
ex.printStackTrace();
}
}
_documentPane.setActiveDocument("Employé");
}

public static void actionNewService() {
if
(!_documentPane.isDocumentOpened("Service"))
{
_documentPane.openDocument(new
DocumentComponent(new NewService(_frame),
"Service", "Service"+ "", new
ImageIcon(_frame.getClass().getResource("/poi
ntage/ressources/mini_service.png"))));
}
_documentPane.setActiveDocument("Service");
}
public static void actionNewStag() {
if
(!_documentPane.isDocumentOpened("Stagiaire")
) {
_documentPane.openDocument(new
DocumentComponent(new NewStag(_frame),
"Stagiaire", "Stagiaire"+ "", new
ImageIcon(_frame.getClass().getResource("/poi
ntage/ressources/mini_stag.png"))));
}
_documentPane.setActiveDocument("Stagiaire");
}
public staticvoidactionNewCommune() {
if
(!_documentPane.isDocumentOpened("Commune"))
{
_documentPane.openDocument(new
DocumentComponent(new NewCommune(_frame),
"Commune", "Commune"+ "", new
ImageIcon(_frame.getClass().getResource("/poi
ntage/ressources/mini_service.png"))));
}
_documentPane.setActiveDocument("Commune");
}
public staticvoidactionNewGrade() {
if (!_documentPane.isDocumentOpened("Grade"))
{
_documentPane.openDocument(new
DocumentComponent(new NewGrade(_frame),
"Grade", "Grade"+ "", new
ImageIcon(_frame.getClass().getResource("/poi
ntage/ressources/mini_service.png"))));
}
_documentPane.setActiveDocument("Grade");
}
public staticvoidactionNewFonction() {
```

```
if
(!_documentPane.isDocumentOpened("Fonction"))
{
_documentPane.openDocument(new
DocumentComponent(new NewFonction(_frame),
"Fonction", "Fonction"+ "", new
ImageIcon(_frame.getClass().getResource("/poi
ntage/ressources/color-line.png"))));
}
_documentPane.setActiveDocument("Fonction");
}
public staticvoidactionNewMatching() {
if
(!_documentPane.isDocumentOpened("Pointage"))
{
_documentPane.openDocument(new
DocumentComponent(new NewMatching(_frame),
"Pointage", "Pointage"+ "", new
ImageIcon(_frame.getClass().getResource("/poi
ntage/ressources/14_layer_novisible.png"))));
}
_documentPane.setActiveDocument("Pointage");
}
public staticvoidactionSetting() {
}
public staticvoidactionStatistic() {
if
(!_documentPane.isDocumentOpened("Statistique
")) {
_documentPane.openDocument(new
DocumentComponent(new Stat(_frame),
"Statistique", "Statistique"+ "", new
ImageIcon(_frame.getClass().getResource("/poi
ntage/ressources/mini_service.png"))));
}
_documentPane.setActiveDocument("Statistique"
);
}
public staticvoidactionPaie() {
if
(!_documentPane.isDocumentOpened("Paiement"))
{
_documentPane.openDocument(new
DocumentComponent(new GestPaie(_frame),
"Paiement", "Paiement"+ "", new
ImageIcon(_frame.getClass().getResource("/poi
ntage/ressources/mini_service.png"))));
}
_documentPane.setActiveDocument("Paiement");
}
public static void actionNewPerson() throws
UnsupportedLookAndFeelException {
```

General Conclusion

We made an automatic pointing system by fingerprint based on the extraction of minutiae. We obtained interesting results which allow a better accuracy for agents' payroll by eliminating calculation errors, bad adjustments when rounded, involuntary overpaid waste of time. It also provides statistics on all of the personnel on the basis of well-defined criteria. The indexing of a database in this application confirms the need for tomorrow's biometrics.

References

Books, articles, lecture's notes

[ABD1] D. Abdelhakim, «Approches bio-inspirées pour la reconnaissance des formes», mémoire de fin d'études, Université MENTOURI DE CONSTANTINE, 2006

[BAT2] J.D. BATUBENGA « Etude d'un système de reconnaissance des formes biométriques Application sur l'identification des empreintes digitales», mémoire de DES d'études, Université de Kinshasa, 2010.

[BOU3] F. BOUGAMOUZA, «Contribution à la reconnaissance automatique de l'écriture manuscrite arabe, application sur les montants littéraux des chèques», thèse de doctorat, 2009

[CHA4] A. CHAARI, « Nouvelle approche d'identification dans les bases de données biométriques basée sur une classification non supervisée », thèse, octobre 2009.

[CHA5] F. CHAFIA, « Un Crypto système à Base de la Biométrie Pour l'Authentification», mémoire de fin d'études, 2010.

[CHA6] B. CHARROUX, A. OSMANI & Y. THIERRY-MIEG, «UML 2 pratique de la modélisation», Edition Pearson 2eme Edition, 2009.

[DJU7] S. DJUNGU, «Cours Génie Logiciel et Construction des Programmes», Unikin, Notes de cours 2010.

[GAB8] J. GABAY & D. GABAY, «UML 2 Analyse et conception», Edition Dunod, 2008.

[MOR9] N. MORIZET, « Reconnaissance Biométrique par Fusion Multimodale du Visage et de l'Iris», thèse, École nationale d'Informatique et Télécommunications de Paris, mars 2009.

[MBU10] E. MBUYI MUKENDI, «Cours de Base de données», Unikin, Notes de cours 2009.

[PER11] F. PERRONNIN et J. DUGELAY, « Introduction à la Biométrie Authentification des Individus par Traitement Audio-Vidéo », Revue Traitement du Signal, volume 19, 2002.

[ROQ12] P. Roques & F. Vallée, «UML en Action», Edition Eyrolles 4eme Edition, 2007.