A1.3.2 Package service

Tableau 7.18 : Liste d'interface pour accéder aux services

Interface
IBitmapService

```
package com.hitechs.commons.resources.service;

import java.awt.Component;
import java.awt.Image;
import java.awt.image.BufferedImage;
import java.awt.image.RenderedImage;
import java.io.FileNotFoundException;
import java.io.IOException;
import javax.imageio.ImageWriteParam;
import javax.imageio.ImageWriter;

public interface IBitmapService {

    String getExtensionImage(short typeImage);

     Image resize(Image source, int width, int height) ;

    BufferedImage imageToBufferedImage(Image image);

    int getPageImage(String path) throws FileNotFoundException, IOException ;

    BufferedImage tiffToBufferedImage(String path) throws FileNotFoundException,
IOException ;

    BufferedImage tiffToBufferedImage(String path,int page) throws FileNotFoundException,
IOException ;

    Image rotate(Image source, double angle) ;

    BufferedImage convertRenderedImage(RenderedImage img,double zox,double zoy);

    Image getImageInComponent(Component component);

    BufferedImage getCaptureScreen(int x, int y, int w, int h);

    BufferedImage getCaptureScreen(javax.swing.JFrame frame);

    BufferedImage getCaptureScreen();

    void saveImage(BufferedImage buf,String path,String tipa);

    ImageWriteParam getHeaderParameter(ImageWriter writer);
```

```
    void saveTiffImage(BufferedImage buf,String path) throws IOException;;

    BufferedImage copyImage(BufferedImage src, int imageType) ;
}
```

IDateService

```
package com.hitechs.commons.resources.service;

import java.util.ArrayList;
import java.util.Date;

public interface IDateService {

    Date getCurrentDay();

    String getCurrentDay(String format);

    int getCurrentDayOfMonth();

    int getCurrentDayOfWeek();

    int getCurrentMonth();

    int getCurrentYear();

    int getCurrentWeek();

    void setCurrentDay();

    String getFormatUser();

    void setFormatUser(String formatUser) ;

    int getDayOfMonth(Date date);

    int getDayOfWeek(Date date);

    int getMonth(Date date);

    int getYear(Date date);

    int getWeek(Date date);

    int getWeekOfMonth(Date date);

    Date stringToDate(String date,String format);

    String dateToString(Date date,String format);
```

```
    boolean isDate(String date,String format);
    Date addDays(Date date,int number);
    Date addMonths(Date date,int number);
    Date addYears(Date date,int number);
    Date getDateByWeek(int year, int week);
    Date getDateByWeek(int year, int week,int dayOfWeek);
    Date getDayInWeek(Date m_date,int dayOfWeek);
    Date getDayZeroHour(Date m_date);
    Date getDayEndOfMonth(int year,int month);
    int getMaxDateMonth(int year,int month);
    long getDayCountIntervale(Date start,Date end);
    int getYearCountIntervale(Date start,Date end);
    int getCountDayWeek(int month,int year,int dayWeek) ;
    String getListDays(Date start,Date end,String format,String separator);
    String getListDaysSql(Date start,Date end,String format);
    ArrayList<Date> getListDays(Date start,Date end);
    Date createDate(int year,int month,int day);
    Date changerHours(Date date,int hour,int minute,int seconde, boolean matin);
    String getDayOfWeek(Date date,int lang);
    String getDayOfWeekShort(Date date,int lang);
    String getDayOfWeek(int daty,int lang);
    String getDayOfWeekShort(int day,int lang);
    String getMonthOfYear(Date date,int lang);
    String getMonthOfYear(int mois,int lang);
    boolean isBissextile(int year) ;
}
```

IFileService

```
package com.hitechs.commons.resources.service;

import com.hitechs.commons.resources.dataUni;
import com.hitechs.commons.resources.file.fileInfo;
import com.hitechs.commons.resources.tableUni;
import com.hitechs.commons.resources.vectorUni;
import java.io.File;
import java.util.ArrayList;
import java.util.List;

public interface IFileService {

    boolean isExist(String path);

    boolean isDirectory(String path);

    void createDirectory(String path);

    String getExtension(String path);

    String getNameWithExtension(String path);

    String getNameWithOutExtension(String path);

    void deleteFile(String path);

    void copyFile(String source, String dest);

    void copyUtils(String source, String dest);

     boolean moveFile(String source, String dest);

    ArrayList <String> readLines(String file);

    ArrayList <Object> readCsv(String file,String separator);

    ArrayList <String> readLineInResource(String path);

    ArrayList <Object> readCsvInResource(String chemin,String separateur);

    String readAllText(String path);

    String readAllTextInResource(String chemin);

    void writeText(String path,String text,boolean append,boolean retour);

    void writeTextLines(String path,ArrayList<String> values,boolean append,boolean retour);
```

```
    void writeCsv(ArrayList <Object> contenu,String file,String separator);

    ArrayList<String> getListDirectory(File path);

    fileInfo getFileInfo(File file);

    ArrayList<fileInfo> getListFileInDirectory(String path);

    boolean copieDirectory(String source, String dest);

    boolean deleteDirectory(String path,boolean withParent);

    List lçoadProperties(String path);

    void saveProperties(ArrayList<vectorUni> list,String path,String name);

    void loadProperties(String path,dataUni data);

    void saveProperties(dataUni data,tableUni table, String path,String name);
}
```

ILDAPService

```
package com.hitechs.commons.resources.service;

import com.hitechs.commons.resources.accountLDAP;
import com.hitechs.commons.resources.ldapServer;
import java.util.ArrayList;

public interface ILDAPService {

    boolean isAll();

    void setAll(boolean all);

    ldapServer getServer();

    void setServer(ldapServer server);

    void init();

    boolean authenticate(String login,String pwd);

    ArrayList<accountLDAP> getAllAccounts();

    ArrayList<accountLDAP> getAllAccounts(String dn);

    ArrayList<accountLDAP> getAccountsByEmail(String mail);

    accountLDAP getAccountByEmail(String mail);
```

```
    accountLDAP getAccountByLogin(String login);

    accountLDAP getAccountByLogin(String dn,String login);

    ArrayList<accountLDAP> getAccountByName(String name);

    ArrayList<accountLDAP> getAccountsByName(String dn,String name);

    ArrayList<accountLDAP> getAllAccountGroup(String group);

    ArrayList<accountLDAP> getAccountsGroup(String group,String name);

    boolean isAccountGroup(String group,String login);

    void addAccount(accountLDAP account);

    void updateAccount(accountLDAP account);

}
```

IMailService

```
package com.hitechs.commons.resources.service;

import com.hitechs.commons.resources.dataMail;
import com.hitechs.commons.resources.headerUni;
import java.util.ArrayList;

public interface IMailService {

    String getMailHost();

    void setMailHost(String mailHost);

    String getPort();

    void setPort(String port);

    boolean isVerbose();

    void setVerbose(boolean verbose);

    boolean isDebug();

    void setDebug(boolean debug);

    boolean isHtml();

    void setHtml(boolean html);
```

```
    String getContentType();

    void setContentType(String contentType);

    dataMail getNewEmail();

    void sendEmail(dataMail mail);

    ArrayList<dataMail>readMails(String address,headerUni criteria);

}
```

IPDFService

```
package com.hitechs.commons.resources.service;

import java.io.File;
import java.io.InputStream;

public interface IPDFService {

    String PDFToText(String path);

    String PDFToText(String path, int startPage,int endPage);

    void mergePDF(String fileName1,String fileName2,String fileName);

    void JpegToPDF(InputStream stream,String fileName);

    void tiffToPDF(File file,String fileName);

    void tiffToPDF(File file,String fileName,int page);

    void PDFToTiff(InputStream stream,String fileName);
}
```

ISheetService

```
package com.hitechs.commons.resources.service;

import com.hitechs.commons.resources.binaryFile;
import com.hitechs.commons.resources.component.grid.metaColumn;
import com.hitechs.commons.resources.office.cellHeader;
import com.hitechs.commons.resources.office.cellInfo;
import com.hitechs.commons.resources.office.rangeInfo;
import com.hitechs.commons.resources.property.UIProperty;
import java.io.InputStream;
import java.util.ArrayList;
public interface ISheetService {
```

```
    void openNewBook();

    void openBook(InputStream stream);

    void openBook(String path);

    void openModelBook(String pathModel,String realPath);

    void createSheet(String name,ArrayList<metaColumn> columns,int column,int
row,ArrayList<UIProperty> properties);

    void createSheet(String name,ArrayList<cellHeader> columns);

    void fillSimpleBorder(int page,int firstRow,int lastRow,int firstCol,int lastCol);

    String getCellValue(int page, int row, int col);

    int getColumnMax(int page);

    int getRowMax(int page);

    int getSheetCount();

    void setColumnWidth(int page,int column,int width);

    ArrayList<Integer> getColumnsWidth(int page);

    void fillSheetPerRow(int page,ArrayList<rangeInfo> liste,ArrayList<UIProperty> properties);

    void fillSheetPerCell(int page,ArrayList<cellInfo> liste,ArrayList<UIProperty> properties);

    void insertImage(binaryFile image,String extension,int page,int column,int row);

    void mergeCells(int page,rangeInfo range);

    void fillRange(int page,rangeInfo range);

    void setStyleRange(int page,rangeInfo range,ArrayList<UIProperty> properties);

    ArrayList<String> readListByRow(int page, int row);

    void saveToFile(String path);

    void saveToPdf(String path);

    void close();
}
```

ITraductorValue

```
package com.hitechs.commons.resources.service;

import com.hitechs.commons.resources.dataUni;

public interface ITraductorValue {

    void setMultipleIn(boolean multipleIn);

    boolean isMultipleIn();

    void addValue(Object code,Object value);

    Object getValue(Object code);

    Object getValue(dataUni data);
}
```

A1.3.3 Package utils

Tableau 7.19 : Liste d'interface aux événements

Interface	**Description**

IEventObject	package com.hitechs.commons.utils;
	import java.util.EventListener;
	
	public interface IEventObject extends EventListener{
	
	public void doEvent(Object obj);
	
	}

IFiredKeyEvent	package com.hitechs.commons.utils;
	
	public interface IFiredKeyEvent {
	
	public void firedKeyEvent(java.awt.event.KeyEvent evt);
	
	}

ANNEXE 2 : BPMN

BPM (Business Process Management) est une discipline qui consiste à considérer la gestion des processus comme un moyen d'améliorer la performance opérationnelle. Les processus métier sont représentés sous forme de modèles graphiques grâce à l'ensemble des conventions graphiques BPMN (BPMN Business Process Model and Notation).

A2.1 Version BPMN 2.0

BPMN 2.0 est un langage de modélisation qui remplace BPMN 1.x et BPEL pour exécuter les processus modélisés.

Il contribue à l'amélioration de la norme BPMN 1.x en apportant :

- un méta modèle normalisé et un format de sérialisation pour BPMN, qui permet aux utilisateurs d'échanger des modèles BPMN entre les outils de différents fournisseurs.
- une sémantique d'exécutions normalisées pour BPMN, qui va permettre aux fournisseurs logiciels d'implémenter des moteurs d'exécution interopérables pour les processus métier.
- un format d'échange graphique, permettant aux utilisateurs d'échanger les informations graphiques d'un diagramme de processus métiers
- une notation étendue pour les interactions inter-organisationnelles (également connues sous le nom de chorégraphies processus), qui permettra de créer de nouveaux cas d'utilisation pour les outils automatisés de soutien pour les processus qui impliquent plusieurs partenaires.
- un processus de transfert détaillé de BPMN pour WS-BPEL, montrant l'alignement de BPMN avec les outils et les normes existants
- certains éléments de modélisation supplémentaires pour des processus tels les événements et sous-processus non-interrompus.

A2.2 Fonctionnement du langage

BPMN est constitué d'un ensemble d'éléments graphiques. Ces éléments permettent le développement de schémas simples. Les éléments ont été choisis pour se distinguer les uns des autres. Par exemple, les activités sont des rectangles, et les décisions sont des diamants. Les différents éléments graphiques de la notation sont organisés selon un concept de layer (couche) extensible. Chaque couche est construite en se basant sur la couche précédente.

A2.3 Élément de notation

La spécification de BPMN 2.0 définit les symboles qui seront utilisés pour créer des modèles de processus. La spécification décrit la syntaxe et la sémantique de tous les symboles. Dans la pratique, les modélisateurs de processus sont libres de choisir des symboles pour modéliser.

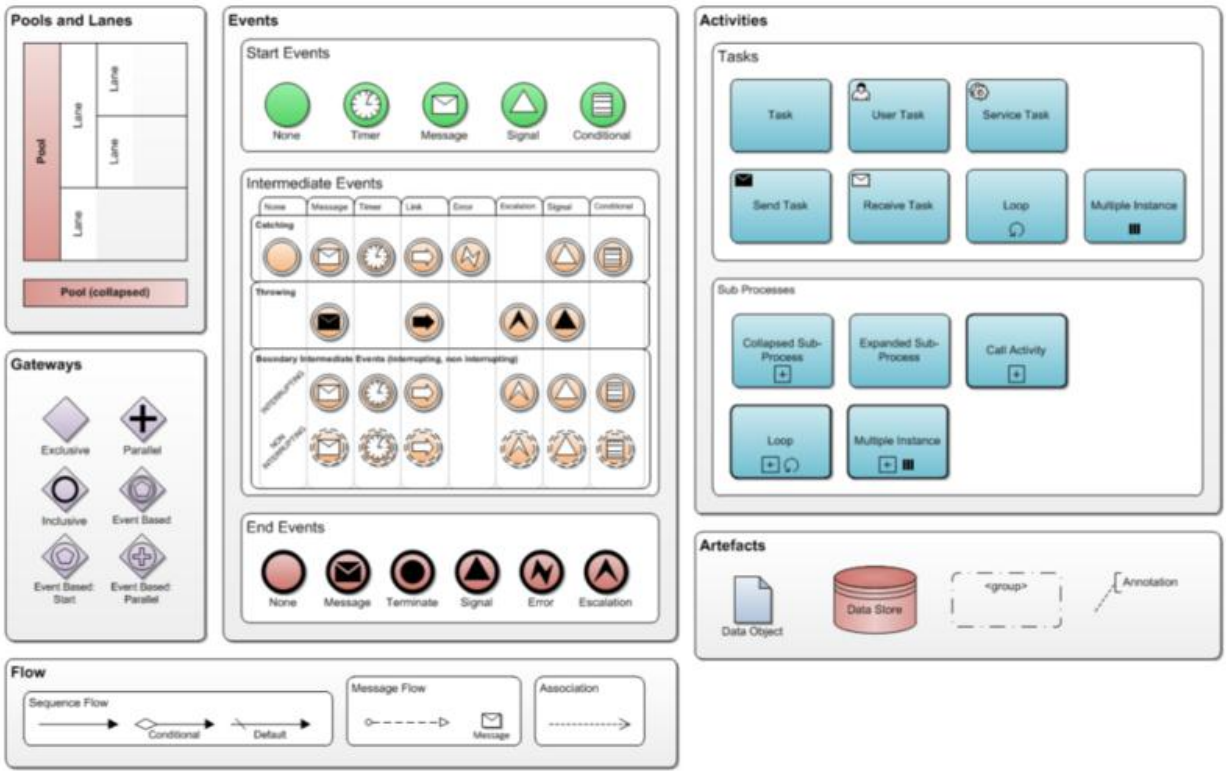

Figure 8.1 : Vue globale des icônes BPMn

La spécification BPMN 2.0 fournit trois sous-ensembles prédéfinis, appelé sous-classe de conformité (conformance sub-classes) :

- descriptif : éléments et attributs nécessaires à la description d'un processus de haut niveau.
- analytique
- exécutables