Computers and Electronics in Agriculture 130 (2016) 1–12

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

A system for the rapid design and implementation of Personalized Agricultural Key Performance Indicators issued from sensor data

Sandro Bimonte [a,*], Echcherki Naoufal [a], Laurent Gineste [b]

[a] UR TSCF, Irstea, 9 Av. Blaise Pascal, 63178 Aubiere, France
[b] Exotic Systems, 29, rue Georges Besse, 63100 Clermont-Ferrand, France

ARTICLE INFO

Article history:
Received 31 May 2016
Received in revised form 9 September 2016
Accepted 17 September 2016
Available online 7 October 2016

ABSTRACT

Farm Management Information Systems provide decision-makers (farmers) with a set of predefined Key Performance Indicators (KPIs) that are issued manually from collected data or via sensors installed on farm equipment. However, agricultural decision-makers need ad-hoc KPIs issued from low-cost sensors.

Therefore, we present our *VBoxReporting* system in this paper. VBoxReporting is a complete system that allows the rapid design and implementation of reporting KPIs issued from VBox data. VBox, which was developed by Exotic Systems, is a low-cost sensor for tracing the energy use of agricultural equipment. Once the indicators are modeled, VBoxReporting automatically generates all of the scripts required to load data from VBox and create the indicators. We also provide a usability study to validate our user interface layer.

1. Introduction

In the era of Spatial Big Data (Shekhar et al., 2012), increasing amounts of geo-referenced data are available via new acquisition data systems (e.g., remote sensing, social networks, sensors) (Lee and Kang, 2015). In this context, GeoBusiness Intelligence (GeoBI) technologies represent first-class citizens of systems that allow the analysis of Spatial Big Data. GeoBI systems include Spatial Data Mining, Spatial OLAP, spatial statistical tools, and reporting systems. Reporting systems (e.g., SpagoBI) allow decision-makers to analyze geo-referenced Key Performance Indicators (KPIs) by using (cartographic) visual displays (Angelaccio et al., 2012; Golfarelli et al., 2013). Those KPIs are usually issued from the data sources stored in existing transactional databases. Reporting tools have been successfully used in several application domains, such as marketing and health, and more recently, in the agricultural context (Leonardi et al., 2014). KPIs have recently been integrated in Farm Management Information Systems (FMISs) (Fountas et al., 2015a). Sørensen et al. (2010) defined an FMIS as a "planned system for collecting, processing, storing, and disseminating data in the form needed to carry out a farm's operations and functions." According to Fountas et al. (2015a, 2015b), FMISs provide functionalities for field operation management, best practice tools, finance, inventory, traceability, reporting, site-specific tools, sales, machinery management, human resource management, and quality assurance. In particular, "traceability" and "site-specific tools" functionalities concern the analysis of KPIs regarding the equipment (e.g., tractors) and inputs (e.g., fuel) that are deployed during agricultural technical operations (e.g., plowing). In particular, the agricultural KPI analysis offered by FMISs generally utilizes sensor data (Fountas et al., 2015b). Therefore, smart farming has been defined as the integration of sensor networks into farms to better manage farm activities. The usage of sensor data in FMISs for smart farming raises several problems, including scalability, real-time integration, and quality (Lee and Kang, 2015). Moreover, as noted by Bimonte (2015), the agricultural context is characterized by its own features, which implies adapting the existing GeoBI technology functionalities. In particular, agricultural decision-makers are usually *non-skilled Information Technology (IT) decision-makers* (Bimonte, 2015) who demand *low cost materials* (Murakami et al., 2007) and *personalized KPIs* (Bimonte et al., 2013). These requirements are not supported by existing systems. Indeed, although GeoBI technologies have proven useful for analyzing the data issued from (mobile) sensors (Lee and Kang, 2015), the rapid design of personalized KPIs from this particular type of data remains unexplored. Existing FMISs and sensor platforms provide a set of predefined KPIs, which might not correspond to the agricultural decision-makers' needs (Bimonte, 2015). Indeed, ETL (Extraction-Transformation-Loading) routines must be implemented to generate indicators from sensor data. These ETL scripts, which can be complex and time-consuming (Kimball, 1996), extract various subsets of data from data sources, transform them using ETL

* Corresponding author.
E-mail addresses: sandro.bimonte@irstea.fr (S. Bimonte), naoufal.echcherki@irstea.fr (E. Naoufal), Laurent.Gineste@exotic-systems.com (L. Gineste).

http://dx.doi.org/10.1016/j.compag.2016.09.010
0168-1699/

functions (e.g., rename, aggregate) or other means, and load them in various other storage systems (DataBase Management Systems). Therefore, when data sources (i.e., sensor data) and KPIs are predefined, implementing a set of predefined ETL routines is possible. Per contra, when indicators are personalized ones, ad-hoc ETL routines must be defined according to particular decision-makers' needs. In this case, these personalized indicators cannot be available as native FMSI KPIs, but they should be created on demand by the IT experts for the end users (i.e., decision-makers) using various ETL tools such as Talend. Although ETL tools are used for this task, we find that ETL systems are too expressive for our needs. Indeed, ETL systems provide generic functionalities to extract and transform data, contrary to our approach, where the extraction of indicators from sensor data can be seen as a simplified ETL model. Moreover, because ETL systems do not provide any automatic mechanisms for the generation of these indicators, the implementation of ETL routines can be time-consuming and error-prone.

Another limitation is that agricultural decision-makers are non-skilled IT decision-makers; hence, conceptual ETL/database models (such as that of El Akkaoui et al. (2013)) are not sufficiently easy and readable to represent an exchange support between IT experts and decision-makers (Bimonte et al., 2013). Consequently, another exchange support, one that is familiar to decision-makers, should be used.

Therefore, *an ad-hoc sensor KPI* (i.e., KPIs issued from sensor data) *methodology and system are needed.* To the best of our knowledge, only the study of Bimonte et al. (2013) has designed OLAP databases supplied with on-demand data from sensors for agricultural decision-makers. However, as underlined in Bimonte et al. (in press), KPIs can be distinguished in three main categories (OLAP indicators, Stream indicators, and OLTP indicators) according to data sources, queries and data navigation. OLTP indicators correspond to reported KPIs (simply named indicators in the rest of the paper) because they are based on transactional data, data navigation is not allowed, and their queries are predefined. This type of KPI allows the definition of graphical dashboards that decision-makers can use to analyze data at a glance.

To conclude, the main goal of this work is to provide a new methodology for the design of smart farm KPIs issued from sensor data. This methodology should allow decision-makers to easily and rapidly conceive KPIs according to particular farmers' analysis needs from data collected using low-cost sensors. These KPIs should be transparently integrated in a classical FMIS to provide cartographic, tabular and graphical display reports.

Therefore, we present our *VBoxReporting* system in this paper. VBoxReporting is a complete system that allows the rapid design and implementation of reporting KPIs issued from VBox data. VBox, which was developed by Exotic Systems, is a low-cost sensor for tracing the energy use of agricultural equipment. Once the indicators are modeled, VBoxReporting automatically generates all of the scripts required to load data from VBox and create the indicators.

In this manner, VBoxReporting allows farmers (i.e., decision-makers) to analyze their agricultural activities by using the KPIs that correspond to their own economical, agricultural and personal management of their farm(s). VBoxReporting frees farmers from the KPI usage imposed by the agricultural equipment manufacturers or FMIS providers.

The paper is organized as follows: requirements for such a solution are detailed in Section 2; the case study is detailed in Section 3; Section 4 describes the main functionalities of the VBoxReporting system; in Section 5, the focus is the design of indicators' user interfaces; the prototyping methodology is detailed in Section 6; various validation experiments are shown in Section 7; related works are presented in Section 8; and a discussion of our proposal is detailed in Section 9.

2. Functional requirements

In this section, we describe the functional requirements of our proposal. Therefore, we present the main features of the agricultural decision-making process based on agricultural KPIs. In this work, we exclusively consider the indicators that can be represented using classical SQL "Aggregation-group by" queries, i.e., the indicators that can be calculated from relational database tables using SQL queries that aggregate numerical attributes using an SQL aggregation function (SUM, MIN, MAX) grouped by some alphanumerical attributes, as shown in the following:

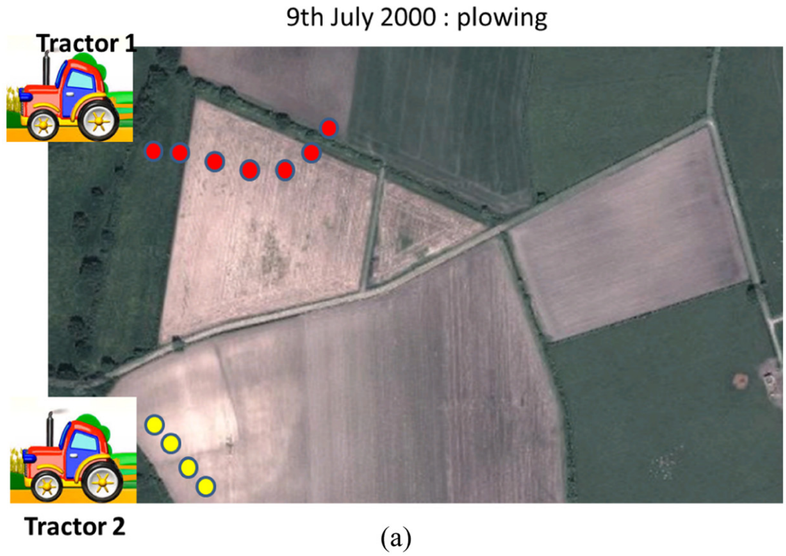
```
Select Aggregation(numerical attribute)
From TABLES
Group by alphanumerical attributes
```

For example, the KPI "*The total consumption of fuel per plot per day*" aggregates via SUM the numerical attribute representing the fuel and groups it by two alphanumeric attributes: plot and day. Its SQL formulation is shown in Fig. 12b.

In fact, the usage of Information Technologies (ITs) by farmers is limited to a small number of applications: weather forecast, economic products and retail. In particular, a recent French study (BVA, 2013) stated that French farmers use specialized websites to plan their activities according to the weather forecast (e.g., MeteoFrance), buy or sell materials (e.g., Leboncoin.fr), and check product prices (e.g., http://www.terre-net.fr/). They also use email for their professional exchanges. Other ITs, such as social media, are mainly used for personal reasons. From a more general point of view, farmers are usually non-skilled ITs users (Bimonte et al., 2013). Then, an IT targeted to farmers should (i) *be as easy to use as possible*, (ii) *provide information via a simple website or emails*, and (iii) *represent indicators via visual means to which farmers are familiar, e.g., bar and pie charts*.

However, there are also some more expert famers. Indeed, agricultural decision-makers also occasionally use Farm Management Information Systems (FMISs) (Fountas et al., 2015a) to manage their exploitations from several points of view, including economic, practice, and administrative. FMISs include all of the above-described tools, as well as various other interesting ones, such as inventory and sales. In particular, FMISs provide users with various analytical functionalities concerning technical operations and sales, among others (Fountas et al., 2015a). These predefined indicators are issued from a set of particular data that the decision-makers can integrate into their FMIS: data coming from a set of particular sensors and data manually inserted by the decision-makers. Moreover, farms can be considered ad-hoc economic systems. Indeed, a farming system is defined as "any level of unit(s) engaged in agricultural production as it is wedded in a social, political, economic, and environmental context" (Turner and Brush, 1987). This implies that each farm needs different decisional indicators depending on the level of the unit: the spatial scale (few or many hectares), the owned resources (e.g., animals, plots), or employees (familiar or enterprise farm). Then, (iv) *a decisional indicator should be conceived according to the needs of each farmer*. In other words, it is difficult for a particular farmer to use a palette of predefined indicators because these indicators might not be adapted to his/her farming system.

Finally, among others, a very recurrent obvious concern for farmers is the return of investment (BVA, 2013). This implies that when ITs are used in their activities, they should be (v) *low-cost solutions* because the most expensive recipes are dedicated to their classical activities (e.g., materials and product purchases) (Murakami et al., 2007).



(a)

message_id bigint	numero_serie character varying(50)	message character varying(256)	date timestamp without time zone
1000008	vbox-001-00002	GPS:1443795701;162141021015;4.5442079E;48.5434526N	2015-10-02 16:21:41.0599
1000009	vbox-001-00003	GPS:1443795701;162141021015;4.5452079E;48.5404526N	2015-10-02 16:21:41.1531
1000010	vbox-001-00004	GPS:1443795702;162142021015;4.5462079E;48.5374526N	2015-10-02 16:21:42.4931
1000011	vbox-001-00004	MEMS:1443795760;0141D4935490813F	2015-10-02 16:21:42.4953
1000012	vbox-001-00004	MEMS:1443795746;0141D4935390013D	2015-10-02 16:21:42.4984
1000013	vbox-001-00004	MEMS:1443795734;0141D4935390013E	2015-10-02 16:21:42.5003
1000014	vbox-001-00004	MEMS:1443795712;0141D4825290014F	2015-10-02 16:21:42.5024
1000015	vbox-001-00004	FUEL:1443795702;521	2015-10-02 16:21:42.5043
1000016	vbox-001-00001	GPS:1443795705;162145021015;4.5432079E;48.5464526N	2015-10-02 16:21:45.7315
1000017	vbox-001-00002	GPS:1443795711;162151021015;4.5444344E;48.5435556N	2015-10-02 16:21:51.0627
1000018	vbox-001-00002	MEMS:1443795718;0141D48A53912059	2015-10-02 16:21:51.0645
1000019	vbox-001-00002	MEMS:1443795763;0124C7A5647D03B9	2015-10-02 16:21:51.0663
1000020	vbox-001-00002	MEMS:1443795716;0141D494528D205A	2015-10-02 16:21:51.0687
1000021	vbox-001-00002	MEMS:1443795756;0141D42D4C8E005B	2015-10-02 16:21:51.0710
1000022	vbox-001-00002	FUEL:1443795711;510	2015-10-02 16:21:51.0737
1000023	vbox-001-00003	GPS:1443795711;162151021015;4.5454344E;48.5405556N	2015-10-02 16:21:51.1556
1000024	vbox-001-00004	GPS:1443795712;162152021015;4.5464344E;48.5375556N	2015-10-02 16:21:52.5069
1000025	vbox-001-00004	SID:1443795792;206B46A400270025;0056; 23.2;3.262	2015-10-02 16:21:52.5087
1000026	vbox-001-00004	SID:1443795798;03003E001100064C;0056; 23.2;3.262	2015-10-02 16:21:52.5101
1000027	vbox-001-00001	GPS:1443795715;162155021015;4.5434344E;48.5465556N	2015-10-02 16:21:55.7332
1000028	vbox-001-00002	GPS:1443795721;162201021015;4.5446623E;48.5436586N	2015-10-02 16:22:01.0820

(b)

id_plot	label_plot	geo_plot
0	HMP	*NULL*
1	P1	0103000000010000000500000002F35423F5347484055F833B...
2	P2	010300000001000000050000004D66BCADF4464840990E9D9...
3	P3	010300000001000000050000009D83674293464840B43D7AC...
4	P4	01030000000100000005000000234A7B832F464840014C193...
5	P5	01030000000100000005000000554E7B4ACE454840B64DF1B...
6	P6	01030000000100000005000000C3B81B446B4548406A32E36...
7	P7	01030000000100000005000000A296E65608454840B2497EC...
8	P8	01030000000100000005000000A08D5C37A54448407D3F355...

(c)

Fig. 1. (a) Case study example, (b) VBox data, Geographic data.

3. Case study

In this section, we present a set of valuable indicators conceived by Exotic Systems, based on their domain knowledge, to describe and validate the proposal. Therefore, four indicators have been proposed to monitor the plot work in terms of energy use and work duration. In particular, we consider 10 tractors equipped with sensors (i.e., VBox; c.f. Section 4.1). Each tractor has associated work equipment, for example, a plow. Each day, some tractors work on the plots of the farm to achieve a particular technical operation, such as plowing (Fig. 1a). The data acquired by the sensors are transferred via GSM or using a USB key to the PC server of the farm.

An example of raw data collected by VBoxes (i.e., sensors) is shown in Fig. 1b. The integration of sensor data with data from decision-makers describing their plots (Fig. 1c) allows the definition of indicators combining these two different data sources.

plot	day	sum_fuel
HMP	2015-09-13	431
P1	2015-09-13	565
P2	2015-09-13	586
P3	2015-09-13	561
P4	2015-09-13	571
P5	2015-09-13	572
P6	2015-09-13	585
P7	2015-09-13	571
P8	2015-09-13	517

Fig. 2. Result of the Indicator I2: The total consumption of fuel per plot per day.

Therefore, some KPI at different spatial and temporal scales have been defined:

I1: *The total consumption of fuel per plot.*
I2: *The total consumption of fuel per plot per day.*
I3: *The total consumption of fuel per SID per day and plot.*
I4: *Work duration per SID, day, plot and technical operation.*

In the rest of the paper, we focus on the indicator I2. Fig. 2 illustrates an excerpt of the resulting tuples for I2. Fig. 2 shows how much fuel is consumed per plot and day. For example, on 13 September 2015, '565' liters of fuel were used for the 'P1' plot. Let us note that these personalized indicators use VBox data, which always have the same structure, and decision-makers' data, such as the plots of the farm. An example of a graphical result is shown in Fig. 9.

4. VBoxReporting

In this section, we describe the global architecture of our VBoxReporting system (Fig. 3) and highlight the main features.

The VBoxReporting multi-tier architecture is composed of: the *VBox* layer, *VBox ETL* layer, *Data* layer, *Spatial ETL* layer, *BI-UI*, *Indicators* layer, and *Reporting* layer. The different layers are detailed in the next sections.

4.1. Sensors

The VBox layer is an automatic data recording system that is embedded in mobile equipment, such as a tractor. This low-cost device integrates a recording unit with a GPS receiver, a digital fuel level sensor, several basic sensors "all or nothing," and a Radio Frequency Identification system with RFID tags associated with each piece of agricultural equipment. The georeferenced data files with the time, the GPS position, the states of sensors and the fuel level saved in the Vbox are exported by means of a USB key or GSM. Therefore, the usage of VBoxes as acquisition data tools allows the support of the "Low-cost solutions" requirement described in Section 2.

The *VBox ETL* layer implements a Spatial ETL using Spatial Data Integrator (SDI, 2015) to load raw VBox data (Fig. 1b), cleans it, and loads it in the databases of the Data layer. A Spatial ETL extends an ETL tool with native support for spatial data management. Indeed, raw VBox data presents a "Message" attribute that contains different information: the GPS position, the identifier of the equipment, and the fuel level. Thus, this table is not directly exploitable in a classical relational DBMS using SQL queries. Therefore, the message attribute values are extracted and stored using a well-defined data structure in the Data layer, as described in the following.

4.2. Data sources

The *Data* layer is in charge of storing data coming from the installed VBoxes and decision-makers' data by using the PostGIS Spatial DBMS (PostGIS, 2015). VBox data are stored in the tables described in Fig. 4. The table "sensor_data" contains the fuel values and 4 foreign keys to 4 tables representing the VBox sensor ("vbox"), the equipment ("equipment"), temporal data ("time"), and the position ("position"). This logical model is inspired by the well-known star schema of data warehouses because it speeds up the join queries (Kimball, 1996).

The 'fuel' attribute contains the fuel gauge level. For example, the vbox '1' sent a message at time '232904' in the position '495256???' with a fuel level equal to '223.' An excerpt of the Vbox_data table tuples is shown in Fig. 5.

Therefore, the spatial and temporal scales of the VBox data are the point and the second, respectively.

Using only VBox data, it is possible to provide decision-makers with indicators that use fuel and MEMS as measures grouped by any temporal level, vbox, point, and equipment, for example, *The total consumption of fuel per vbox*, *The total consumption of fuel per vbox and day* or *The total consumption of fuel per vbox*, *day*, *and sid*. Indeed, these indicators can be expressed using simple SQL "Aggregation group-by" queries, as stated in Section 2. For

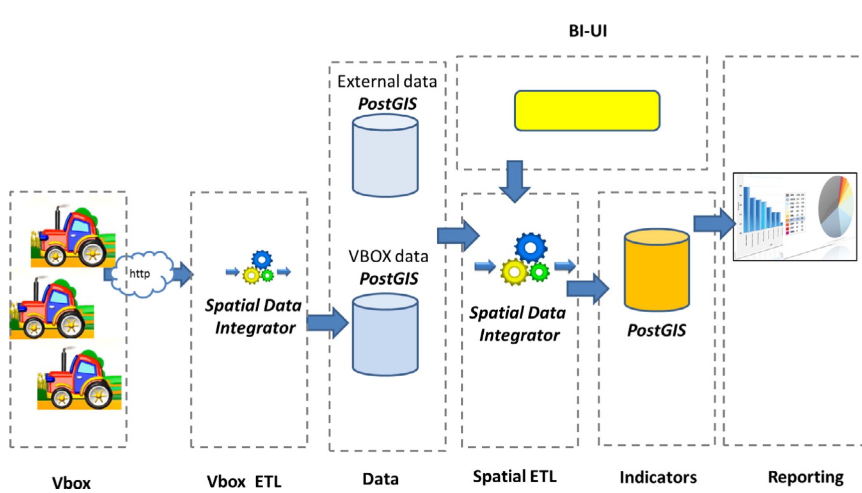


Fig. 3. VBoxReporting architecture.

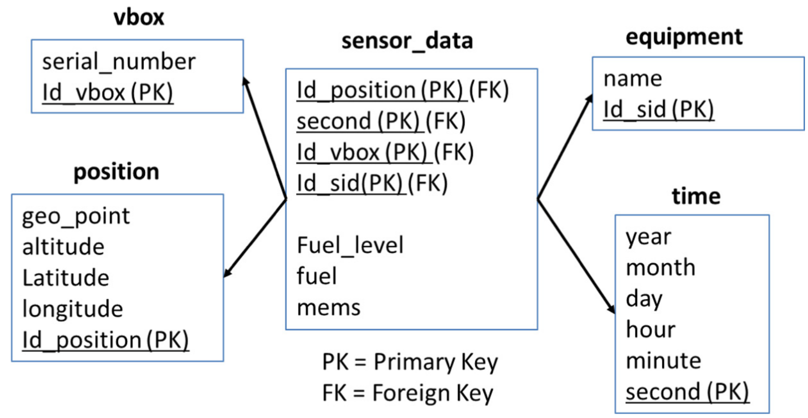


Fig. 4. Data layer: VBox data.

id_data	id_vbox	id_time	id_position	id_sid	fuel
316931	4	123159	274424	44063	215
316901	4	123144	274394	44055	218
316875	4	123127	274371	44055	220
486431	3	195901	419779	66501	221
486430	3	195901	419779	66502	221
316851	4	123105	274340	44047	222
316852	4	123105	274340	44048	222
350832	4	138440	303486	48548	223
575558	1	232904	495256	78111	223

Fig. 5. An extract of the table 'vbox_data'.

```
<?xml version="1.0" encoding="UTF-8"?>
<!--La(es)table(s) de(s) reference(s) à intégrer-->
- <Ref>
      <!--Vous pouvez definir plusieurs tables à intégrées -->
  - <Table PK="id_plot" name="plots">
        <!--name: nom de la table de référence. PK: nom de la colonne
      <Column name="id_plot" additional="y" type="integer"/>
        <!--name: nom d'une colonne. type: type de données -->
      <Column name="name" additional="n" type="string"/>
      <Column name="geom" additional="n" type="geometry"/>
    </Table>
</Ref>
```

Fig. 6. XML representation of the external data of the table of Fig. 1c.

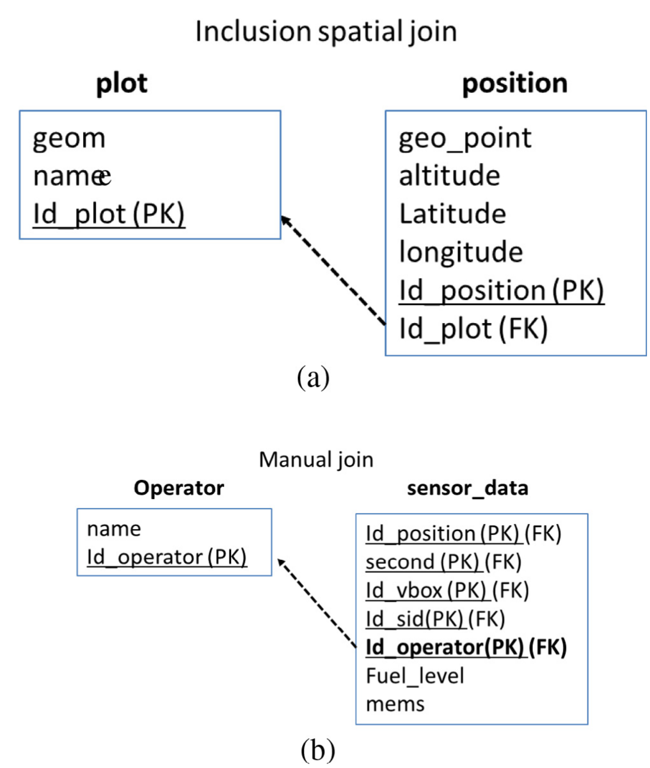


Fig. 7. Adding external data to VBox data.

example, "*The total consumption of fuel per vbox and day*" can be represented as

```
Select SUM(fuel), serial_number, day
From sensors_data, vbox, time
Where sensors_data.id_vbox=vbox.id_vbox and sen
  sors_data.second=time.second
Group by day, serial_number
```

These indicators use only data collected by the Vboxes

As stated in the introduction, decision-makers need to have personalized indicators that use their own data. To solve this issue, external data that are useful for the definition of the indicators (see Section 6) are also stored in the Data layer. For that goal, we propose describing each decision-maker table using an XML file.

For example, indicator I2 needs the geometries and the names of the plots of the decision-maker. Then, a table will be stored (Fig. 1c) with the associated XML file (Fig. 6). The XML describes each attribute of the table with its type.

The external data XML description is necessary because it will be used by the other layers for the automatic generation of the ETL scripts and by the BI_UI user interface.

Two different possibilities are offered to the users to use external data for the indicator design. The first is to add a coarser level to the analysis axes (vbox, time, position and equipment – Fig. 4) of the VBox data. For example, adding the plot level to the indicator (*The total consumption of fuel per plot*) means defining a spatial join between the table position and the table representing the plots (external data) (Fig 7a).

Therefore, the SQL representation for I1 is

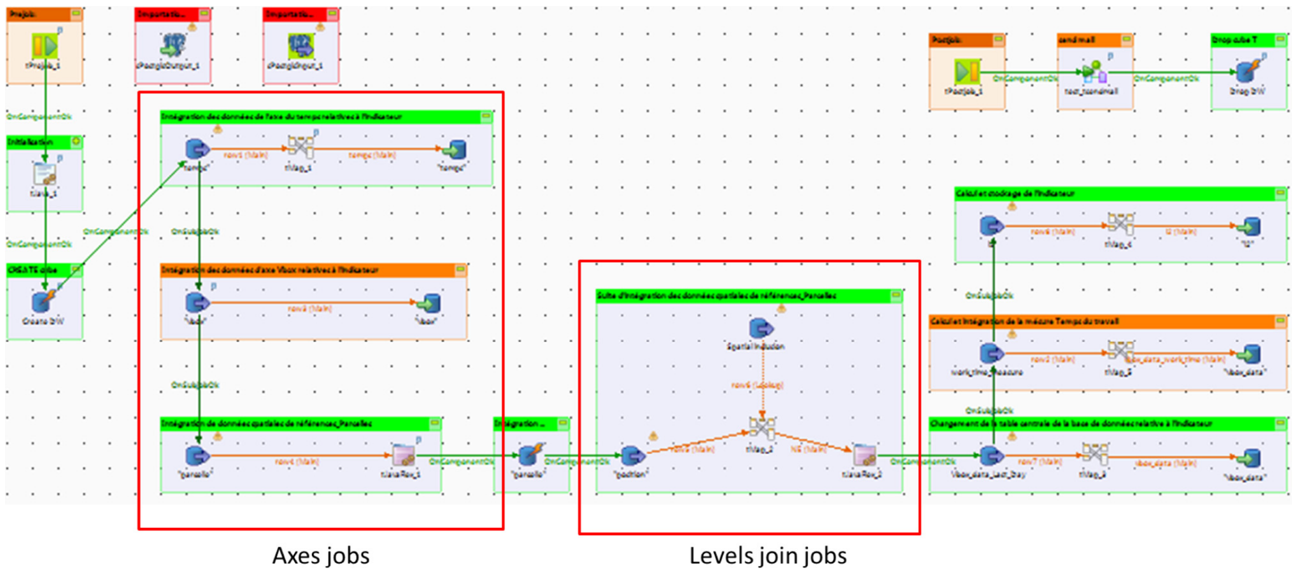


Fig. 8. Spatial ETL template.

```
Select SUM(fuel), plot
From sensors_data, position, plot
Where sensors_data.id_position=position.id_posi
  tion and plot.id_plot=position.id_plot
Group by day, serial_number
```

Moreover, the decision-makers can add a new analysis axis to the VBox data, for example, to analyze fuel by operator (I1: *The total consumption of fuel per plot, and operator*). In this case, a manual join should be defined between the sensor_data table and the external table representing the operators (Fig 7b.)

In this manner, **our proposal allows the use of both sensor data and external data to design personalized indicators** (requirement: "*a decisional indicator should be conceived according to the needs of each farmer*").

4.3. Indicator computation

The *Spatial ETL* layer is in charge of extracting data from the *Data* layer, calculating the indicator results and storing them in the Indicators data layer. It is implemented using SDI. The *Indicators* layer stores the results of the indicators in various PostGIS tables. An example is the table shown Fig. 1.

Because each indicator is different and cannot be defined a priori, considering external data can be used for its definition, the Spatial ETL routines are different. However, to make this phase as automatic as possible, we have conceived a generic ETL template (Fig. 8) that is fulfilled using scripts generated by the BI-UI layer.

From Fig. 8, we can note two customizable groups of ETL jobs: axes jobs and level join jobs. An axes job adds a dimension table to the "sensor_data" table. A level join adds a level to a dimension table. The other jobs are predefined. Jobs on the left create the databases, jobs on the right create measure values, and jobs on the top are in charge of creating the report and sending its pdf file by email.

Although an automatic approach is feasible for our system, in this work, we have chosen a semi-automatic approach for the ETL for two main reasons: the visual approach and extensibility.

Using an ETL tool, IT experts have a visual framework for configuring the ETL routines with all of its functionalities. Therefore, as widely recognized, a visual representation of ETL jobs eases the configuration and development of IT experts.

Using a semi-automatic approach grants the modularity of the ETL routines. This implies a simpler debugging, personalization, and implementation process because the total ETL is not encoded in a unique program code that is difficult to read and customize. Moreover, a semi-automatic approach allows the ETL scripts to be personalized to account for, e.g., new aggregation functions and methods that are not supported at this step by the BI-UI layer.

4.4. Visualization

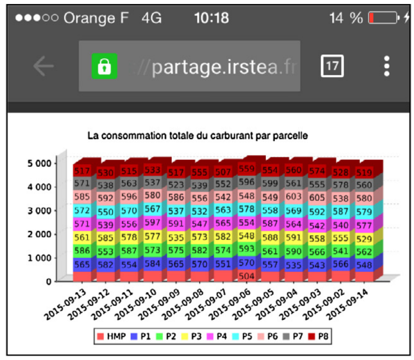
The *Reporting* layer is implemented using JasperReports (JasperReports, 2015). Reports are generated by using data stored in the Indicators data layer and sent by email in pdf format via SDI (requirement: "*provide information via a simple website or email*"). Fig. 9a shows an example of visualization of the pdf report of the Indicator I2 on a smartphone. We have chosen this mobile support and graphical display because they are familiar to agricultural decision-makers, as described in Section 2 (requirement: "*indicators should be represented using a visual means to which they are familiar, such as bar and pie charts.*"). JasperReports defines reports using a particular XML file, where the data source and the report's graphics are defined. An example is shown in Fig. 9b. This XML file is also automatically generated by the BI-UI layer.

Finally, the *BI-UI* layer is responsible for the automatic generation of SQL ETL scripts of the *Spatial ETL* and *Indicators* layers, as described in the next section. Moreover, we have implemented a more sophisticated reporting visualization that has been tested (Fig. 9b) to validate the feasibility of integrating our tool in a classical FMIS.

5. VBoxReporting BI-UI

In this section, we present the core of the VBoxReporting system: the *BI-UI* layer. The *BI-UI* layer is a simple "point and click" user interface. It allows the easy creation of the indicators in just a few steps. A demo video of the user interface is available here: https://www.youtube.com/watch?v=Ga-2SJsMu7w&feature=youtu.be.

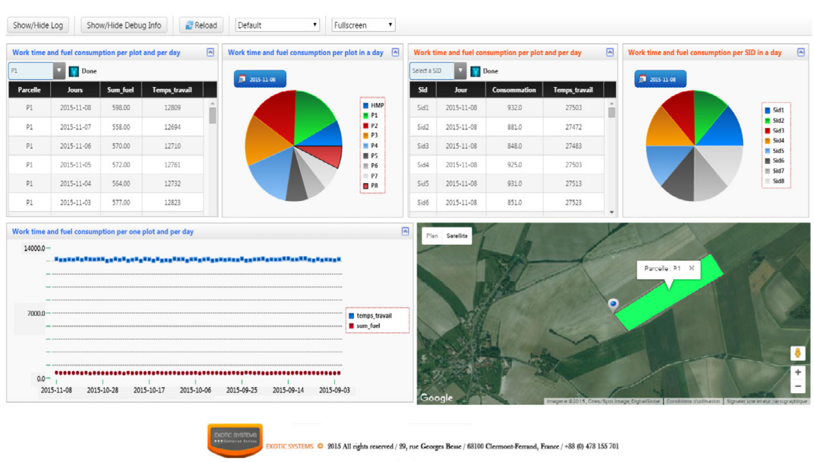
The first step consists of defining joins between the "sensor_data" table and external data tables stored in the *Data* layer, as



(a)

```
- <queryString>
      <![CDATA[select * from i1 order by day ASC]]>
  </queryString>
  <field name="day" class="java.lang.Timestamp"/>
  <field name="name" class="java.lang.String"/>
  <field name="sum_fuel" class="java.lang.Double"/>
+ <background>
+ <title>
+ <pageHeader>
+ <columnHeader>
+ <detail>
+ <columnFooter>
+ <pageFooter>
- <summary>
   - <band splitType="Stretch" height="413">
      + <staticText>
      - <stackedBar3DChart>
         - <chart theme="aegean" renderType="draw">
              <reportElement height="383" width="572" y="30" x="0" backcolor="#FFCC99" mode="Opaque"/>
              <chartTitle/>
              <chartSubtitle/>
            + <chartLegend backgroundColor="#FFFFFF">
           </chart>
         + <categoryDataset>
         + <bar3DPlot isShowLabels="true">
        </stackedBar3DChart>
     </band>
  </summary>
</jasperReport>
```

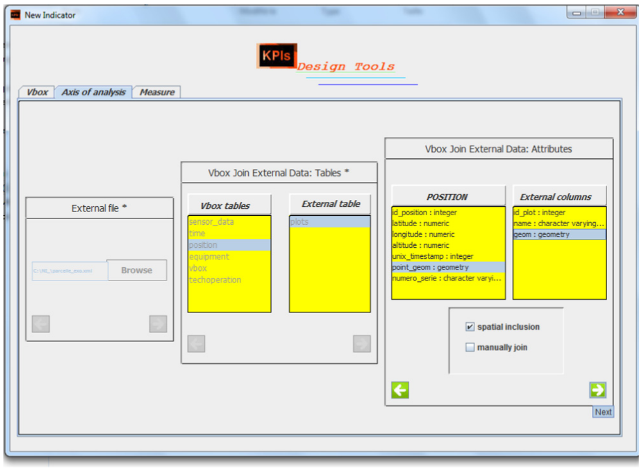
(b)



(c)

Fig. 9. (a) Mobile reporting via Email, and (b) Website reporting.

described in Section 4.2. Therefore, it automatically loads the XML files of the *Data* layer. Then, it prompts the IT expert to define a join between the VBox tables (Fig. 4) and the loaded tables (plotted in our example) using the "VBox Join External Data: Tables" panel (Fig. 10a). Then, the IT expert chooses the attributes to define the join predicate in the two selected tables ("VBox Join External Data: Attributes" panel). In Fig. 10a, she/he defines a join predicate between the "point" attribute of the "position" table and the "geom" attribute of the "plots" table. When the chosen attributes are spatial ones, a spatial inclusion predicate is proposed to the IT expert. In Fig. 10a, the spatial inclusion predicate is used to associate each VBox acquisition data point with the plot that



(a)

```
SELECT id_position, plot.id_plot
FROM position LEFT OUTER JOIN plot
ON ST_CONTAINS (plot.geo_plot,
position.geo_point);
```

(b)

Fig. 10. External data: (a) Visual interface and(b) Automatically generated SQL.

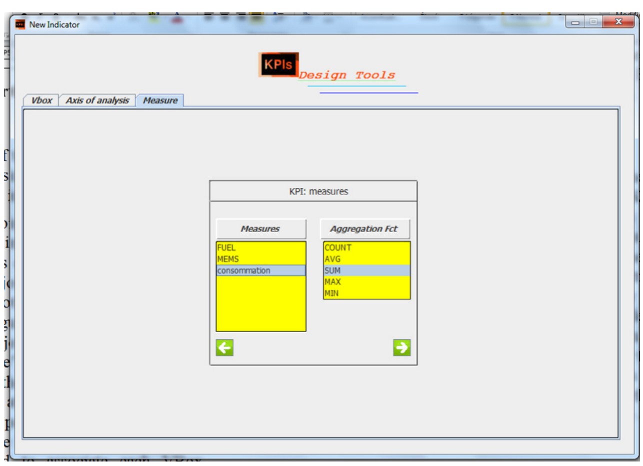


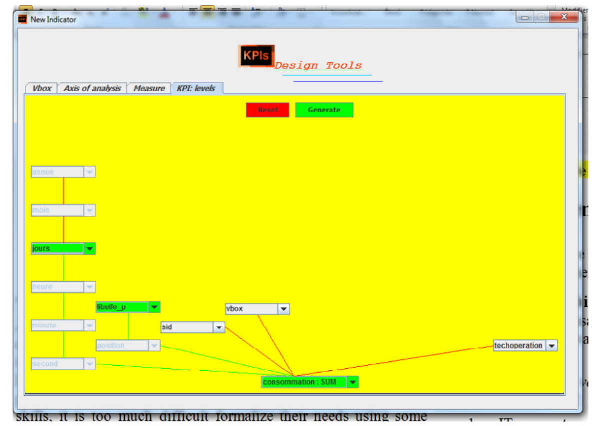
Fig. 11. Measures and aggregation functions.

topologically contains its GPS value (Fig. 1a). Then, the *BI-UI* layer automatically generates the SQL scripts for the Spatial ETL layer, as shown in Fig. 10b for our spatial join example. This operation is repeated for each external data table.

In this way, external data of decision-makers can be easily associated with "sensor_data" to design indicators using VBox and external data.

Once all of the data are available, the indicator can be designed using the user interface.

Therefore, the following step is the choice of the numerical attribute value of the "sensor_data" table to use as the KPI. The "Measures" panel allows IT experts to choose among a set of attributes and a set of SQL aggregation functions. For example, in Fig. 11, she/he chooses the sum of the "consummation" attribute. This step allows the "Aggregation (numerical attribute)" statement of the SQL query representing the KPI to be defined (for example, the first line of the SQL query of Fig. 12b).



(a)

```
SELECT plot.plot,time.day,SUM(sum_fuel)
FROM vbox_data, time, plot, position
WHERE vbox_data.id_time=time.id_time
      AND vbox_data.id_position=position.id_position
      AND position.id_plot=plot.id_plot
GROUP BY plot.plot, time.day
```

(b)

Fig. 12. Analysis axes levels.

Then, the last step consists of the choice of the spatial, temporal, and thematic granularities for the chosen KPI. It is simply achieved with the "Indicator" panel. It presents to the IT expert a set of predefined temporal scales (second, minute, hour, day, month, and year). It also presents the attributes of the external data (i.e., the plot and position for our example) and the attributes of the "equipment" and "sensor_data" tables. Therefore, by simply selecting one or more of these attributes, the indicator scales are defined. In our example of Fig. 12a, the IT expert decides to analyze the sum of fuel per day and plot (the attributes of the other tables are not selected). This step allows the generation of the "group by" statement of the SQL query representing the KPI (for example, the last line of the SQL query of Fig. 10).

Finally, the SQL query implementing this indicator is automatically generated (Fig. 12b). The results of this query are stored in the *Indicators* layer and presented to the decision-makers via the *Reporting* layer.

6. VBoxReporting methodology

As stated in (Bimonte et al., 2013), a prototyping methodology is needed when decision-makers are not skilled IT users. In particular, as identified in (Bimonte et al., 2013), to validate the implemented KPIs, they need to "see" the obtained results using simple graphical displays. Indeed, because farmers are usually not IT skilled users, it is too difficult for them formalize their needs using some software engineering tool or methodology or exchange with IT experts by means of conceptual models (e.g., UML). However, when the resulting indicators can be shown to us, the IT experts' implementations can be understood and validated. Therefore, a prototyping methodology that allows the rapid and easy implementation of those indicators is needed because it allows the drastic reduction of the duration of one design step, the overall number of design steps and, thus, the duration of the entire project (Torlone, 2003). This also implies an important economic gain for IT companies, who can consequently offer low-cost solutions to farmers.

According to the above-stated issues, we propose a prototyping methodology (Fig. 13) based on the VBoxReporting system.

The steps of the methodology are as follows:

(1) Decision-makers informally define their indicators.
(2) Decision-makers provide IT experts with their external data.
(3) IT experts using the BI-UI layer design the indicator.
(4) Raw VBox data are simulated.
(5) ETL scripts are automatically generated.
(6) ETL scripts are inserted in the ETL template.
(7) Reports are presented to the decision-makers that validate them (or not).
(8) If the indicators are validated, then VBoxReporting is applied on real data.
(9) If the indicators are validated, go to step 1.

The proposed methodology is inspired from the OLAP indicator described by Bimonte et al. (2013) and has been successfully validated in several real projects.

7. Experiments

In this section, we evaluate the usability of the BI-UI layer.

To validate the usability of the BI-UI layer, we define a set of experiments to evaluate the accuracy and the satisfaction of its usage (Dix et al., 2003).

The *independent variable* used to control the experiments is the *BI-UI's skill level*:

1. IT experts involved in the VBoxReporting project development (denoted as E). These users know very well how the BI-UI layer works.
2. IT experts involved in the VBoxReporting project. These users are trained by E users (denoted as NE). They do not have knowledge about the underlying mechanisms of the project.

In our experiments, we involve 2 members of the VBoxReporting project and three 1-h-trained NE persons of the Irstea's COPAIN team without any previous knowledge about this project.

The experiment was comprised of the 4 indicators previously described because they are representative of real KPIs and are structurally different from the level points of view.

The two dependent variables that the experiments are meant to evaluate are (Dix et al., 2003):

1. User accuracy and
2. User satisfaction.

User accuracy is a quantitative measurement of the user's ability to solve problems with a specific tool, and it represents the degree of efficacy of use. User satisfaction evaluates whether a user is encouraged to use the tool.

User satisfaction is evaluated with a questionnaire submitted to the users. It contains the following questions:

1. *Did you feel comfortable in performing the most complex task* (design of the indicator I4)? (answers: *definitely yes, yes, unsure, no or definitely no*).
2. *What are the main drawbacks you experienced?* (open answer).

The hypotheses that we want to confirm with these experiments are:

1. There are no differences in accuracy for user skill levels.
2. There are no differences in accuracy for task achievement (KPI design).

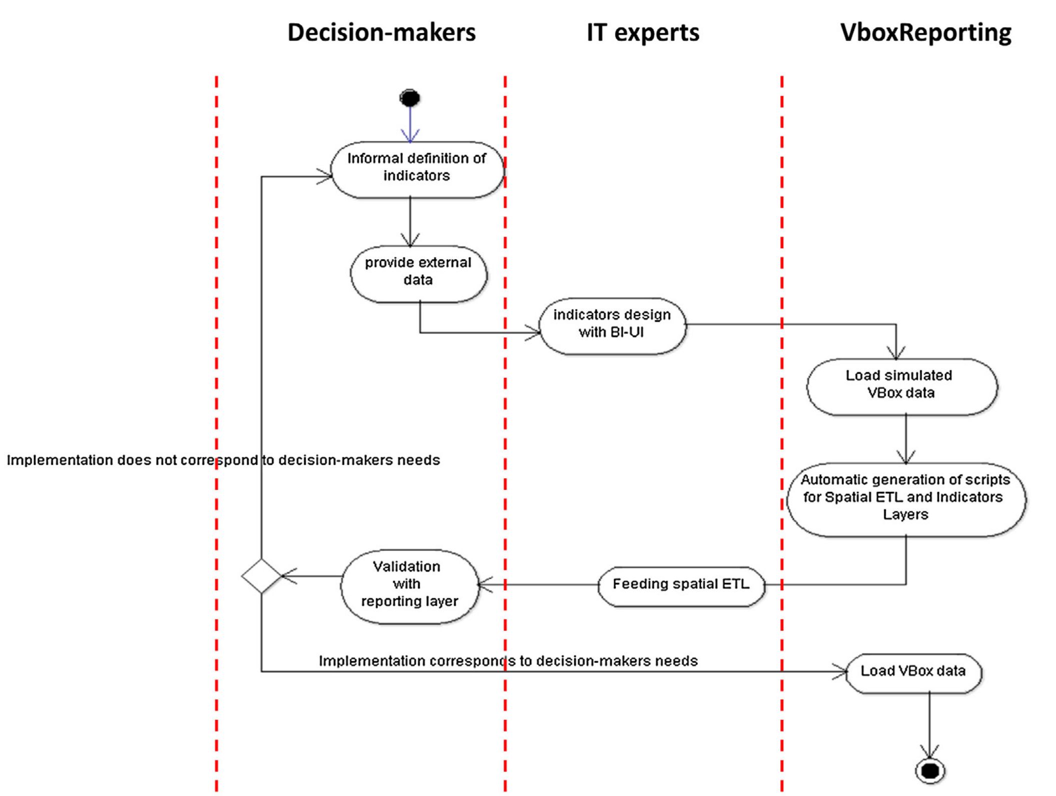


Fig. 13. VBoxReporting methodology.

Note that time performances are not detailed because they are 5 min on average.

We have classified the errors in three main groups according to the usage of the user interface. Our user interface is based on the "Data flow" metaphor (Tjan et al., 1993). Each step (i.e., panel) of the user interface is associated with a particular data operation: data integration, aggregation definition, and level definition. Consequently, these steps are natively organized as an ordered actions flow. Then, we classify the errors to fulfill the flow order:

1. *Low:* the error can be resolved by resting in the current step. For example, at the data integration step (Fig. 10), if the IT expert selects an incorrect format file, he/she is informed by the user interface, and he/she can change it by choosing the right one.
2. *Avg:* the error can be resolved by going back to the previous step. For example, at the level definition step (Fig. 12), if the IT expert wants to change the measure, he/she has to return to the aggregation definition step (Fig. 11).
3. *High:* the error can be resolved by going back to the first step. For example, at the level definition step (Fig. 12), if the IT expert wants to change the levels, he/she has to return to the data integration step (Fig. 10).

Details of experiments for one user are shown in Table 1.

Table 2 shows the percentage of Low errors per users (for all indicators). We can note that the number of errors different from Low error is very small per user, and it does not significantly vary based on expert and non-expert users. Then, the hypothesis "There are no differences in accuracy for user skill levels" is confirmed by our experiments.

Table 3 shows the percentage of Low errors per indicator (all users). As in the previous table, the standard deviation is very low, and the hypothesis "There are no differences in accuracy for task achievement (KPI design)" is confirmed by our experiments.

For the User Satisfaction to the query "*Did you feel comfortable in performing the most complex task* (design of the indicator I4)?" All users answered *definitely yes.*

Some common answers to the question "What are the main drawbacks you experienced?" are as follows:

- A description of attribute tables will be appreciated.
- The user interface should suggest a unique indicator name.
- It should be nice to remove the first step.

To conclude, from these experiments, we can state that the BI-UI is usable and is well adapted for its task: the design of indicators.

8. Related work

Smart farming introduces sensor data for the analysis and the management of farm activities. In fact, several works have focused on new architectures based on cloud infrastructures (Kaloxylos et al., 2014; Mocanu et al., 2015) for the collection, storage and visualization of agricultural data issued from sensor networks. Mocanu et al. (2015) proposed a cloud-based architecture for farm management. Humidity, temperature and soil composition data are integrated in a local repository with other operational data, and then they are integrated with another local repository in the cloud, increasing the quality of the products grown. These new FMISs also provide new methods for visualizing the temporal (Bojan et al., 2015) and spatial components (Serrouch et al., 2015) of sensor data. Moreover, in addition to storing and handling historical data, smart farm systems provide integrated alert systems with intelligent data control mechanisms (Raducu et al., 2015). Other solutions based on cloud databases (such as NoSQL systems) for storing sensor data in real time have recently been investigated (van der Veen et al., 2012; Damaiyanti et al., 2014). Although important recent literature on architectures, databases and decision-support systems for agricultural big data is emerging (Karmas et al., 2016), the design of smart farm KPIs has not been greatly explored because it concerns low-cost sensors, KPI design, and ETL models. Therefore, in this section, we detail the existing work on these three topics.

The existing solutions for data acquisition are expensive and difficult to implement on old equipment. In addition, these solutions do not acquire data frequently enough (hours, days) to allow detailed analyses of the tasks performed by the equipment (Chanet et al., 2005). It is also often necessary to supplement the automatic acquisitions with manual data entry, which is tedious and error-prone. Therefore, we have developed a fully automatic data acquisition system with an event data acquisition trigger: when an equipment actuator changes state, a georeferenced acquisition with a time stamp is made. In this manner, we can follow all of the equipment activity.

Three approaches for the design of KPIs have been developed. The first uses conceptual models based on UML. Bimonte et al. (in press) defined a UML profile allowing the conceptual representation of different types of indicators. However, they did not support an automatic implementation. Palpanas et al. (2007) defined a UML profile for defining the indicators and their visualization. They use the Model Driven Architecture, where the implementation is automatically derived from the conceptual model. However, the ETL routines are not represented in this proposal. The other approach is based on the usage of sophisticated visual languages. Several visual languages for defining queries on top of well-structured spatial databases have been defined (Bonhomme et al., 1999). The main limitations of this approach are: it is difficult to express complex "join-group by" queries, and ETL routines are not supported. Finally, the last approach consists of ad-hoc reporting tools. These tools use advanced "point and click" user interfaces and allow the design of complex generic indicators. However, data preparation is usually done in another tool (ETL tool), or when they present some ETL functionalities, they are very complex to use because they try to be as generic as possible.

Finally, the three approaches described above have also been applied to ETL. For example several works propose conceptual

Table 1
Detailed user accuracy errors of User NE1.

Indicator I1: total fuel consumption per parcel per day/user	Performance		Errors	
Step/Panel	Time	Number of clicks	Description	Level
Initialization/vbox	40	11	Bad choice of XML file input for vbox schema	Low
Enrichment an axis/axis analysis	46	16	Bad join: join a level to a fact table	Low
Selecting the measure/KPI measures	9	3		
Selecting levels/KPI levels	13	5	Choosing the name of a table instead of an attribute	Low
Generation of SQL	16	1		
Total	124	36		

Table 2
User accuracy errors per user.

User accuracy errors per user and type of error		Low	Avg	High	Percentage low errors
E	E1	2	0	0	1
	E2	1	1	0	0.5
NE	NE1	4	1	2	0.571428571
	NE2	2	1	1	0.5
	NE3	3	2	0	0.6
Standard deviation					0.051507875

Table 3
User accuracy errors per indicator.

User accuracy errors per indicator and type of error	Low	Avg	High	Percentage low errors
I1	5	0	0	1
I2	2	2	2	0.333333333
I3	1	2	2	0.2
I4	1	2	2	0.2
Standard deviation	0.382970843			

models for ETL models (El Akkaoui et al., 2013; Wilkinson et al., 2010; Deufemia et al., 2014). Recent work has proposed a visual language for ETL (Caruccio et al., 2014b) based on sketches (Caruccio et al., 2014a). In the same manner, several ETL tools exist, such as Talend. They present a user interface, which allows the definition of all possible ETL operations. These works all present the same limitation: they are very complex to use because they try to be as generic as possible, and they are not exclusively focused on the design of particular indicators.

9. Discussion

In this section, we discuss the main advantages and limits of our approach.

As underlined above, our VBoxReporting methodology allows farmers to conceive personalized KPIs issued from low-cost sensor data and their own data. The proposed methodology is a prototyping design methodology because it allows rapid implementation of the KPIs by means of a few iterations, as shown in Section 8. Indeed, with VBoxReporting, it is possible to design a complex KPI in a short amount of time because the implementations of the complex underlying ETL routines are completely automated. Therefore, the adoption of a prototyping methodology that integrates farmers' data allows KPIs that reflect farmers' analysis needs to be conceived. Moreover, our system allows the integration of such generated KPIs in any existing FMIS because our implementation of the ETL routines is fully based on standard languages, such as SQL and XML. Therefore, VBoxReporting allows farmers (i.e., decision-makers) to be free to decide what, when, and how to analyze their data, without being restricted to generic KPIs defined by providers of sensors and/or FMISs.

Although VBoxReporting seems to support all of the functional requirements defined in Section 2, it does present some limits.

VBoxReporting is exclusively based on the usage of Relational DBMSs for the storage and the analysis of data. When KPIs are needed in real time and the historical data are not necessary, Relational DBMSs do not seem to be efficient due the important time-consuming operations of insertion and delete used by VBoxReporting. In the same way, when KPIs require very huge volumes of historical data, Relational DBMSs do not appear to be the best solution to handle ETL and analytical queries. Another important limitation is the manual feed of ETL jobs by computer science experts because this phase requires important ETL skills and is not generic for all ETL tools.

10. Conclusion and future work

Smart farming has been defined as the integration of sensor networks into farms to better manage farm activities. Therefore, Farm Management Information Systems provide decision-makers (farmers) with a set of predefined Key Performance Indicators (KPI) are issued from manually collected data or via sensors installed on farm equipment. This work is motivated by the lack of an FMIS supporting the following requirements: (i) *be as easy to use as possible*, (ii) *provide information via a simple website or email*, (iii) *represent indicators via visual means to which farmers are familiar, such as bar and pie charts*, (iv) *create decisional indicators according to the needs of each farmer*, and (v) *focus on low-cost solutions*.

Therefore, we present our *VBoxReporting* system in this paper. VBoxReporting is a complete system that allows the rapid design and implementation of reporting KPIs issued from VBox data. VBox, which was developed by Exotic Systems, is a low-cost sensor for tracing the energy use of agricultural equipment. Once the indicators are modeled, VBoxReporting automatically generates all of the scripts required to load data from VBox and create the indicators. VBoxReporting satisfies the technical and functional requirements described above. In particular, to allow decision-makers to provide personalized KPIs, we propose a prototyping methodology. This methodology allows decision-makers to create (and easily implement in an FMIS) indicators according to the particular farmer and farm. To validate the agile characteristics of our methodology, we have conducted some usability studies.

Our ongoing work is the study of different systems for the storage and the analysis of KPIs. For example, we are investigating the usage of Data Stream Management Systems (DSMSs) for alert-type KPIs (Plazas et al., 2016). Indeed, DSMSs support continuous queries in memory-stored data. We are also investigating the usage of NoSQL systems, such as MongoDB or Hbase, for KPIs using huge volumes of data because NoSQL systems grant scalability and high computational performance (Van der Veen et al., 2012).

Our future work is the deployment of VBoxReporting on several farms to validate the scaling of our storage systems and, consequently, usability tests at larger scales. In particular, we plan to compare the design and the implementation of KPIs with and without VBoxReporting. Moreover, we plan to integrate a data-driven methodology to our KPI design framework to attempt to (semi)automatically derive (relaxed) functional dependencies between sensor and external data (under several forms, e.g., multimedia, Linked Open Data, relational data) (Caruccio et al., 2016). This will allow farmers to enlarge their source data and, thus, their analysis capabilities to, for example, compare their farm activities according to meteorological data.

Acknowledgment

The authors would like to thank the "Conseil Régional d'Auvergne" and the European Regional Development Funds (ERDF), which have financially supported this research.

References

Angelaccio, M., Basili, A., Buttarazzi, B., Liguori, Walter. Using geo-business intelligence to improve quality of life. In: Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference, 2–5 October 2012.

Bimonte, Sandro, 2015. Spatial OLAP for agri-environmental data and analysis: lessons learned. MIPRO, 1393–1398.

Bimonte, Sandro, Edoh-Alove, Élodie, Nazih, Hassan, Kang, Myoung-Ah, Rizzi, Stefano, 2013. ProtOLAP: rapid OLAP prototyping with on-demand data supply. DOLAP, 61–66.

Bimonte, S., Schneider, M., Boussaid, O., in press. Business Intelligence Indicators: Types. Models and Implementation, IJDWM (in press).

Bojan, Valentina-Camelia, Raducu, Ionut-Gabriel, Pop, Florin, Mocanu, Mariana, 2015. Cloud-based service for time series analysis and visualisation in Farm Management System. In: Intelligent Computer Communication and Processing (ICCP), 2015 IEEE International Conference on. IEEE, pp. 425–432.

Bonhomme, Christine, Trépied, Claude, Aufaure-Portier, Marie-Aude, Laurini, Robert, 1999. A visual language for querying spatio-temporal databases. ACM-GIS, 34–39.

BVA – TICAGRI. Etude Agrinautes – Agrisurfeurs 2013, Equipements et usages des agriculteurs sur internet.

Caruccio, Loredana, Deufemia, Vincenzo, Polese, Giuseppe, 2014a. A sketch-based conceptual level data integration methodology. IRI, 741–748.

Caruccio, Loredana, Deufemia, Vincenzo, Polese, Giuseppe, 2014b. Visual data integration based on description logic reasoning. IDEAS, 19–28.

Caruccio, Loredana, Deufemia, Vincenzo, Polese, Giuseppe, 2016. Relaxed functional dependencies – a survey of approaches. IEEE Trans. Knowl. Data Eng. 28 (1), 147–165.

Chanet, J.P., Boffety, D., André, G., Humbert, T., Rameau, P., Amamra, A., De Sousa, G., Piron, E., Hou, K.M., Vigier, F., 2005. Wireless Technologies for Field Data Acquisition. In: EFITA 2005. Vila Real, Portugal, p. 6.

Damaiyanti, Titus Irma, Imawan, Ardi, Kwon, Joonho, 2014. Extracting trends of traffic congestion using a NoSQL database. BDCloud, 209–213.

Deufemia, Vincenzo, Giordano, Massimiliano, Polese, Giuseppe, Tortora, Genoveffa, 2014. A visual language-based system for extraction-transformation-loading development. Pract. Exper. 44 (12), 1417–1440.

Dix, Alan, Finlay, Janet E., Abowd, G., Beale, R., 2003. Human-Computer Interaction. Prentice-Hall, Inc..

El Akkaoui, Zineb, Zimányi, Esteban, Mazón, Jose-Norberto, Trujillo, Juan, 2013. A BPMN-based design and maintenance framework for ETL processes. IJDWM 9 (3), 46–72.

Fountasa, S., Carlib, G., Sørensenc, C.G., Tsiropoulosd, Z., Cavalarisd, C., Vatsanidoud, A., Liakosd, B., Canavarie, M., Wiebensohnf, J., Tisseryeg, B., 2015a. Farm management information systems: current situation and future perspectives. Comput. Electron. Agric. 115, 40–50.

Fountas, S., Sorensen, C.G., Tsiropoulos, Z., Cavalaris, C., Liakos, V., Gemtos, T., 2015b. Farm machinery management information system. Comput. Electron. Agric. 110, 131–138. <https://talend-spatial.github.io/>, <http://postgis.net/>.

Golfarelli, Matteo, Mantovani, Marco, Ravaldi, Federico, Rizzi, Stefano, 2013. Lily: a geo-enhanced library for location intelligence. DaWaK, 72–83.

Kaloxylos, Alexandros, Groumas, Aggelos, Sarris, Vassilis, Katsikas, Lampros, Magdalinos, Panagis, Antoniou, Eleni, Politopoulou, Zoi, Wolfert, Sjaak, Brewster, Christopher, Eigenmann, Robert, Terol, Carlos Maestre, 2014. A cloud-based farm management system: architecture and implementation. Comput. Electron. Agric. 100, 168–179.

Karmas, A., Tzotsos, A., Karantzalos, K., 2016. Big geospatial data for environmental and agricultural applications. In: Yu, Guo (Ed.), Big Data Concepts, Theories and Applications. Springer.

Kimball, R., 1996. The Data Warehouse Toolkit: Practical Techniques for Building Dimensional Data Warehouses. John Wiley & Sons, New York. <http://community.jaspersoft.com/project/jasperreports-library>.

Lee, J.-G., Kang, M., 2015. Geospatial big data: challenges and opportunities. Big Data Res. 2 (2), 74–81.

Leonardi, Luca, Orlando, Salvatore, Raffaetà, Alessandra, Roncato, Alessandro, Silvestri, Claudio, Andrienko, Gennady L., Andrienko, Natalia V., 2014. A general framework for trajectory data warehousing and visual OLAP. GeoInformatica 18 (2), 273–312.

Mocanu, Mariana, Cristea, Valentin, Negru, Catalin, Pop, Florin, 2015. Cloud-based architecture for farm management. In: 20th International Conference on Control Systems and Computer Science. IEEE, p. 814–819.

Murakami, Edson, Saraiva, Antonio M., Ribeiro Junior, Luiz C.M., Cugnasca, Carlos E., Hirakawa, Andre R., Correa, Pedro L.P., 2007. An infrastructure for the development of distributed service-oriented information systems for precision agriculture. Comput. Electron. Agric. 58 (1), 37–48.

Palpanas, T., Chowdhary, P., Mihaila, G., Pinel, F., 2007. Integrated model-driven dashboard development. Inform. Syst. Front. 9 (2–3), 195–208.

Plazas, Julián Eduardo, Rojas, Juan Sebastián, Corrales, David Camilo, Corrales, Juan Carlos, 2016. Validation of coffee rust warnings based on complex event processing. ICCSA (4), 684–699.

PostGIS, 2015. Available from: <http://postgis.refractions.net/>.

Raducu, Ionut-Gabriel, Bojan, Valentina-Camelia, Pop, Florin, Mocanu, Mariana, Cristea, Valentin, 2015. Real-time alert service for cyber-infrastructure environments. In: 3PGCIC, pp. 296–303.

SDI, 2015. Available from: <http://www.spatialdataintegrator.com/>.

Serrouch, Adil, Mocanu, Mariana, Pop, Florin, 2015. Soil management services in CLUeFARM. ISPDC, 204–209.

Shekhar, Shashi, Gunturi, Viswanath, Evans, Michael R., Yang, KwangSoo, 2012. Spatial big-data challenges intersecting mobility and cloud computing. MobiDE, 1–6.

Sørensen, G.C., Fountas, S., Nash, E., Pesonen, L., Bochtis, D., Pedersen, S.M., Basso, B., Blackmore, S.B., 2010. Conceptual model of a future farm management information system. Comput. Electron. Agric. 72, 37–47.

Tjan, B., Breslow, L., Dogru, S., Rajan, V., Rieck, K., Slagle, J., Polaic, M. A data-flow graphical interface for querying a scientific database. In: Proceedings of IEEE Symposium on Visual Languages (VL'93), Bergen, Norway, 1993, pp. 49–54.

Torlone, Riccardo, 2003. Conceptual Multidimensional Models. Multidimension. Databases, 69–90.

Turner, B.L., Brush, S.B., 1987. Comparative Farming Systems. The Guildford Press, New York, NY, USA.

van der Veen, Jan Sipke, van der Waaij, Bram, Meijer, Robert J., 2012. Sensor data storage performance: SQL or NoSQL, physical or virtual. In: IEEE CLOUD, pp. 431–438.

Wilkinson, Kevin, Simitsis, Alkis, Castellanos, Malú, Dayal, Umeshwar, 2010. Leveraging business process models for ETL design. ER, 15–30.