Original scientific paper
10.7251/AGRENG1803064K
UDC 636:004.6

DEVELOPMENT OF THE DATA WAREHOUSE ARCHITECTURE FOR PROCESSING AND ANALYSIS OF THE RAW PIG PRODUCTION DATA

Vitalijs KOMASILOVS, Armands KVIESIS, Aleksejs ZACEPINS, Nikolajs BUMANIS*

Department of Computer Systems, Faculty of Information Technologies, Latvia University of Life Sciences and Technology, Jelgava, Latvia
*Corresponding author: nikolajs.bumanis@llu.lv

ABSTRACT

Precision Livestock Farming (PLF) approach requires immense amount of data collection, aggregation and processing, using various hardware and software solutions, and is applied in many farms trying to achieve the most efficient and effective way of production. Hardware and software compatible systems capable of achieving this goal are called Farm Management Information Systems (FMIS), and are a necessity for a complete and successful implementation for Precision Agriculture (PA) branch approaches. However, most of commercially available FMIS do not only focus on crop management, but also have limited availability to small and average sized farms, in terms of price, supported language and specific features. Simpler FMIS, on the other hand, do not have necessary capabilities to fully support PLF. There are currently very small amount of high grade pig farm oriented FMIS, especially for farms with less than a hundred of sows. Therefore, there exists the need for solutions for managing farms with limited number of sows. To help address this need, authors proposed and developed architecture for unified data warehouse (DW), which was scalable and extendable cloud based data storage and processing system with support of individual data analysis. DW has capabilities to data interexchange and/or be integrated in existing FMIS throughout variety of data-in/data-out interfaces, like UIs, unmanned data supplier or consumer systems. The core of the DW is designed to provide data processing flexibility and versatility, whereas data flow within the core is organized between data vaults in a controllable and reliable way.

Keywords: *Data warehouse, information systems, pig farm management, precision livestock farming.*

INTRODUCTION

Currently the maximum efficiency and productivity of a grain or an animal farm can only be achieved by applying technological approach called Precision Agriculture (PA), as well as its branch approaches (Pierpaoli *et al.*, 2013). These approaches allow implementation of methods that not only decrease cost of production, but also increase yields (Far and Rezaei-Moghaddam, 2018). Animal farms fall under Precision Livestock Farming (PLF) branch that is aimed to improve efficiency of production, while also increasing animal and human welfare (Banhazi *et al.*, 2012), for instance, beekeeping (Zacepins *et al.*, 2015, 2016). PLF collects and aggregates large amount of data using monitoring hardware. These data are then processed by usage of sound and/or video labelling and analysis procedures that are in most cases integrated into online automated tools capable of controlling, monitoring and modelling the behaviour of animals and their biological responses (Tullo, 2005; Nasirahmadi, Edwards and Sturm, 2017). Implementation of these automated tools, along with extended functionality, as a management software is called Farm Management Information System (FMIS).

Additionally to monitoring tools, most of commercially available FMIS provide dashboards, reporting and analysis tools, growing and feed management tools, financial management and task planning tools. Overall scope of features depends on sophistication of an offered FMIS. Most FMIS (AgWorld, FarmLogs, FarmWorks, for example) offer mainly crop management oriented tools, some (AgriWebb, for example) provide additional tools for animal management. There are also multiple highly sophisticated and feature rich FMIS like AgritecSoft and AgroSoftLtd. In the recent years development of FMIS mainly focused on online features and cloud capabilities (Welte *et al.*, 2013; Fountas *et al.*, 2015; Ampatzidis *et al.*, 2016), introducing FMIS built around different modules; therefore such FMIS, like CloudFarms, for example, are becoming highly sought after. Cloud and module based FMIS is also most appropriate system for small pig farms, as it allows to fine-tune functionality by choosing combinations of modules according to farm owner's needs. Thus not only providing required functionality, but also giving capabilities to manage overall cost of FMIS.

Despite various efforts taken by the researchers and developers to create user friendly systems for data analysis, there's still a lack of a unified, customizable and flexible systems for the pig sector. It is hard to find a universal system, which would be able to operate with different data inputs and would have flexible data processing option. Authors aim to develop an architecture for system that can be considered as decision making tool with easy and fast data entry (manual or automatic) powered by flexible and detailed reporting.

There are multiple researches aimed to differentiate requirements for pig farm FMIS (Zoranović and Novković, 2013; Husemann and Novković, 2014), including functions, modules and hardware. FMIS architecture as well has it's requirements stated in multiple researches (Murakami *et al.*, 2007; Nikkilä, Seilonen and Koskinen, 2010), that point to the need for one main centralized system, used as a hub or gateway that processes data coming from monitoring hardware. These data are then aggregated and used to create different kinds of reports, available for analysis. One of the implication of such hub is Data Warehouse (DW).

DW in its essence acts as an intermediary between data providers and data consumers (Inmon, 2002; Inmon, B, 2010), and provides customizable facilities for data storage management, processing, analysis and output. The DW is used to help agricultural specialists run the farm more effectively. The ability to manage and effectively present the volume of data tracked in today's agriculture is the cornerstone of data warehousing. But when the data warehouse is replenished in real-time it empowers users by providing them with the most up-to-date information possible. It is possible to create capable FMIS based on DW that can be used in a combination with a Decision Support System in compliance with technical and user requirements. Authors suggest implementing Livestock Object DW as a cloud based data storage and processing unit with capabilities to combine different data sources like existing systems and available on-farm generated data. The proposed platform follows best practices in distributed and asynchronous data processing by utilizing multi-agent techniques in conjunction with real-time data warehousing approach. The aim of this paper is to describe proposed platform's architecture and functionality principles.

MATERIAL AND METHODS

The concept of proposed DW architecture (Figure 1) consists of various main components to ensure that information is stored for further analysis. Input interface provides *data-in* functionality from various sources – it can be in a form of a data file, a measurement system (configured accordingly to send data directly into DW), or third-party services etc. In most cases simple data file gives user a capability to manually upload it. Acceptable file formats are usually not heavy restricted as most systems are able to parse almost all of them. After parsing is done user should specify necessary information regarding metadata.

Data Vault modelling structure provides functionality to track history of the data flow, for instance, sender's credentials, data source and recipient. Data marts, however, are like databases (Casters, Bouman and Van Dongen, 2010) and contains a summarized information (Krneta, Jovanovic and Marjanovic, 2016) ready to be provided through a corresponding output interface.

Data are processed almost immediately after they are received at the DW. Various models are involved to ensure correct data aggregation and processing procedures that are controlled by the DW Core. DW Core is also responsible for reliable data flow management. The data are at first put into a temporary storage called Swamp and are directed to the appropriate vaults afterwards. Data flow is organized by using internal messaging service.

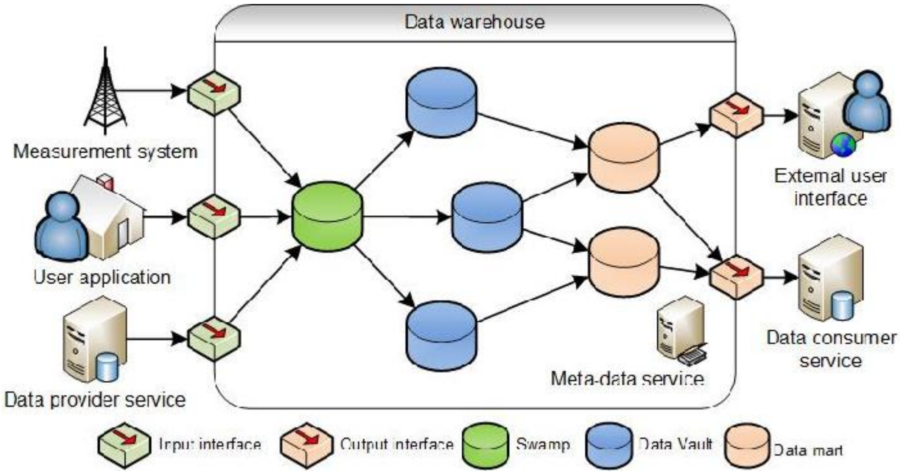

Figure 1. The concept of proposed DW architecture for data processing

Described architecture was implemented in a prototype containing several independent subsystems: DW core for handling data processing, and Web application providing graphical interface for users to submit data for processing and receive results. System was built using technologies such as Spring Boot 2.0 framework, MongoDB database to form backend and Angular 6, Bootstrap 4 framework for frontend functionality.

Results and Discussion

Overall architecture is divided into two main components – connectivity and DW Core. Connectivity corresponds to external systems' connection to DW, and includes such modules as In/Out Interfaces, which are able to use graphic user interface, and Web API that is responsible for first step authorization and verification. Each individual external system is connected to DW core using dedicated integration of Web API, where multiple Web APIs are used to connect to DW core's DataIn and DataOut services.

DW Core that provides routing and processing functionality is built using data input/output and messaging services, temporary and long term databases and processing units. DW Core's architecture contains multiple instances of Vaults, some of which are used as Marts. In proposed DW architecture Vault is a modified version of Data Vault, and differs from traditional understanding of its architecture with integrated calculation processing unit. Each Vault provides self-sufficient and logically independent transformation function, for instance, average weight per week calculation; therefore, as various raw data could be needed for a single final calculation, multiple vaults can be interconnected to build chains of data transformations resulting in flexible and extendible reporting pipeline. Vaults are responsible for processing data according to particular array of tags, such as "pigs", "weight", "week" etc.; whereas data exchange procedures are implemented in a secure way provided by DataIn and Messaging services. DW Core includes multiple databases – one temporary storage that is used to store raw data, called Swamp, and multiple long term storages for processed data and reports, called Repository, one per each Vault.

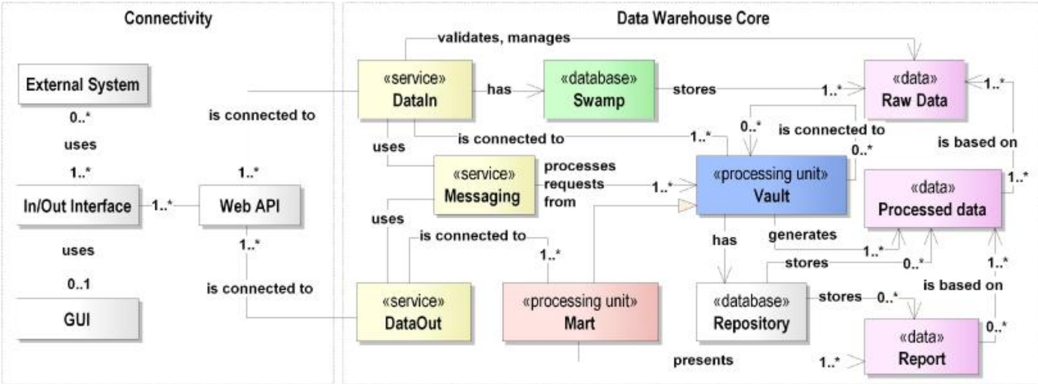

Figure 2. Proposed DW architecture and interaction between components

Proposed DW architecture was approbated using an example of manual data input and processing of these data using three vaults, one of which fulfilled the role of the Mart that was responsible for processed data aggregation and presentation in the form of report.
Data processing flow (Figure 3. Data processing flowFigure 3, [x] – step number) starts with manual data input, where user accesses one of the external systems and uploads a file, containing particular data.

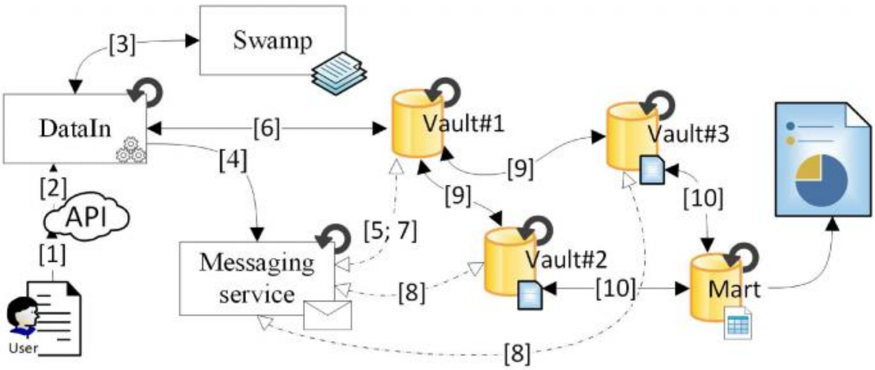

Figure 3. Data processing flow

In this instance [1], user uploaded a file containing pig weight values per day. This file is then processed by Web API that performs authorization based on user's credentials, external system's credentials and data tags included in the file. In case of successful authorization data from the file are converted to JSON format by Web API, and is forwarded [2] to DataIn service for further processing. DataIn service performs validation and verification procedures, and upon successful acceptance transfers data [3] to the Swamp that stores all raw data. As new data are accepted, DataIn service informs [4] Messaging service about its receipt. Messaging service is responsible for informing various processing units that process these particular raw data; in this instance, Messaging service informs [5] Vault#1, that the Swamp now contains data that, based on its data tags, must be processed by this Vault. For security purposes Vaults are not connected to the Swamp; therefore Vault#1 requests these data from [6] DataIn service, that upon successful acceptance transfers requested data to Vault#1, and deletes these data from the Swamp. In proposed system, only one Vault can get and store data from the Swamp; therefore, in this instance, Vault#1 is used only as a hub that contains raw data needed by Vault#2 and Vault#3. After data receipt, Vault#1 informs [7] Messaging service about containing data with particular data tags. Corresponding Vaults are linked together using legal agreement, and receive notification [8] about data availability from Messaging service by usage of Message Queue technique. Vault#2 and Vault#3 take data [9] from Vault#1 required for different calculation functions. In this instance, Vault#2 calculates average values, and Vault#3 – maximum. Whenever there are enough processed data for reporting, Mart takes [10] these data from Vault#2 and/or Vault#3 accordingly to create a report that can be requested from DataOut service by an external system (e.g GUI) on demand and is presented to user in any acceptable format (e.g. graph, chart, table, etc). Raw data and final report resulted from processing these data, including line graphs, is shown in Figure 4.

Raw data

```
day,valA,valB
1,"14,2073549240395","86,4934815692002"
2,"22,5464871341284","8,51390049160926"
3,"20,7056000402993","85,9302380445746"
4,"11,2159875234604","62,931644014674"
5,"13,2053786266843","95,1343886828755"
6,"10,6029225090054","70,868013523287"
7,"22,2849329935939","20,1422573338288"
8,"16,9467912331169","78,7047723552386"
9,"22,0605924262088","52,12051276909"
10,"12,2798944455532","146,962278568105"
11,"9,00004896724648","68,4424779450117"
12,"9,31713540999782","45,2893933749933"
13,"14,1008351841332","88,8161882291309"
14,"18,9530367784744","163,732531611415"
15,"21,2514392007856","67,0667753546281"
16,"13,5604834166747","91,2623410058675"
17,"6,19301254060222","130,79685950695"
18,"10,2450637661416","24,8993892989545"
19,"18,7493860483148","108,749359594869"
20,"20,5647262536381","44,8063512004177"
21,"17,1832781926803","79,233180550269"
22,"17,955743453548","98,3222449183872"
23,"6,76889797912415","103,217123542739"
24,"8,47210818996688","45,0245306972214"
25,"12,3382412495111","137,704557698144"
26,"15,812792252398","40,2744764069246"
```

Processed data and report

Week	Weight (avg)	Weight (max)
1	16.396	22.546
2	14.665	22.061
3	15.392	21.251
4	13.355	17.956
5	15.855	23.000
6	16.068	24.819
7	14.529	23.255
8	15.222	20.351
9	14.874	20.964
10	14.019	24.134
11	14.801	24.998

Export

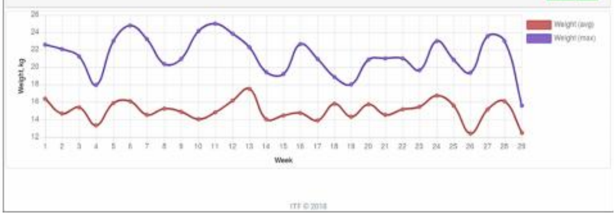

Figure 4. Results of approbated DW architecture

As seen in Figure 4, raw data, containing pig weight values per day, are processed and two calculations are made – average pig weight and maximum pig weight per week. These calculated data are then represented in a form of two line graphs. Different output interfaces are considered, for instance, system allows exporting this report as separate pdf file, as well as accessing it from user-friendly online GUI, connected to DW through DataOut service and Out Interface. This report can also be sent to other external services and/or systems.

CONCLUSIONS

Proposed and developed architecture is universal as it allows connectivity to different multiple external systems, each of which can produce different data aimed for various livestock of PLF branches, for example, cow/pig farming, beekeeping, etc. Implemented platform provides infrastructure for data processing; however business and required calculation logic is defined by PLF industry experts and animal farm owners. Platform provides capabilities to deal with partial data, and specific cases like uploading data file with overlapping or missing data points.

ACKNOWLEDGEMENT

Scientific research, publication and presentation are supported by the ERA-Net SusAn Project „Improving pig system performance through a whole system approach (PigSys - 2817ERA08D)" funded by the Latvian State Education Development Agency, subordinated to the Ministry of Education and Science.

REFERENCES

Ampatzidis, Y. *et al.* (2016) 'Cloud-based harvest management information system for hand-harvested specialty crops', *Computers and Electronics in Agriculture*. Elsevier B.V., 122, pp. 161–167. doi: 10.1016/j.compag.2016.01.032.

Banhazi, T. *et al.* (2012) 'Precision Livestock Farming: An international review of scientific and commercial aspects', *International Journal of Agricultural and Biological Engineering*, 5, pp. 1–9.

Casters, M., Bouman, R. and Van Dongen, J. (2010) *Pentaho Kettle solutions: building open source ETL solutions with Pentaho Data Integration*. John Wiley & Sons.

Far, S. T. and Rezaei-Moghaddam, K. (2018) 'Impacts of the precision agricultural technologies in Iran: An analysis experts' perception & their determinants', *Information Processing in Agriculture*. China Agricultural University, 5(1), pp. 173–184. doi: 10.1016/j.inpa.2017.09.001.

Fountas, S. *et al.* (2015) 'Farm management information systems: Current situation and future perspectives', *Computers and Electronics in Agriculture*. Elsevier B.V., 115, pp. 40–50. doi: 10.1016/j.compag.2015.05.011.

Husemann, C. and Novković, N. (2014) 'Farm Management Information Systems: A Case Study on a German Multifunctional Farm', *Economics of Agriculture*, 61(2), pp. 441–453.

Inmon, B, H. (2010) 'Data Warehousing 2 . 0 Modeling and Metadata Strategies for Next Generation Architectures', *Architecture*, p. 13.

Inmon, W. H. (2002) *Building the data warehouse, John Wiley & Sons, Inc.*

Krneta, D., Jovanovic, V. and Marjanovic, Z. (2016) 'An Approach to Data Mart Design from a Data Vault', *INFOTEH-Jahorina BiH*, 15.

Murakami, E. *et al*. (2007) 'An infrastructure for the development of distributed service-oriented information systems for precision agriculture', *Computers and Electronics in Agriculture*, 58(1), pp. 37–48. doi: 10.1016/j.compag.2006.12.010.

Nasirahmadi, A., Edwards, S. A. and Sturm, B. (2017) 'Implementation of machine vision for detecting behaviour of cattle and pigs', *Livestock Science*. Elsevier B.V., 202(May), pp. 25–38. doi: 10.1016/j.livsci.2017.05.014.

Nikkilä, R., Seilonen, I. and Koskinen, K. (2010) 'Software architecture for farm management information systems in precision agriculture', *Computers and Electronics in Agriculture*, 70(2), pp. 328–336. doi: 10.1016/j.compag.2009.08.013.

Pierpaoli, E. *et al*. (2013) 'Drivers of Precision Agriculture Technologies Adoption: A Literature Review', *Procedia Technology*. Elsevier B.V., 8(Haicta), pp. 61–69. doi: 10.1016/j.protcy.2013.11.010.

Tullo, E. (2005) 'Precision livestock farming: an overview of image and sound labelling', *Electronic Publishing*, 1(1), pp. 30–38. Available at: https://air.unimi.it/retrieve/handle/2434/228259/289284/Paper_Tullo.pdf.

Welte, J. *et al*. (2013) 'An Approach to Farm Management Information Systems Using Task-Specific , Collaborative Mobile Apps and Cloud Storage Services', *American Society of Agricultural and Biological Engineers Annual International Meeting*, 1, pp. 1–23. doi: 10.13031/aim.20131579954.

Zacepins, A. *et al*. (2015) 'Challenges in the development of Precision Beekeeping', *Biosystems Engineering*. Elsevier, 130, pp. 60–71.

Zacepins, A. *et al*. (2016) 'Beekeeping in the future—Smart apiary management', in *Carpathian Control Conference (ICCC), 2016 17th International*, pp. 808–812.

Zoranović, T. and Novković, N. (2013) 'Information System for Small Pig Farm', *Procedia Technology*, 8(0), pp. 544–549. doi: http://dx.doi.org/10.1016/j.protcy.2013.11.076.