Available online at www.sciencedirect.com
ScienceDirect
Computers and Electronics in Agriculture 58 (2007) 37–48
Computers and electronics in agriculture
www.elsevier.com/locate/compag

An infrastructure for the development of distributed service-oriented information systems for precision agriculture

Edson Murakami, Antonio M. Saraiva*, Luiz C.M. Ribeiro Junior, Carlos E. Cugnasca, Andre R. Hirakawa, Pedro L.P. Correa

Agricultural Automation Laboratory, Escola Politécnica da Universidade de São Paulo, Computer and Digital Systems Engineering Department, São Paulo, Caixa Postal 61548, CEP 05424-970, SP, Brazil

Received 15 July 2006; received in revised form 15 October 2006; accepted 11 December 2006

Abstract

Precision agriculture (PA) involves using electronic technology to collect a large amount of data in the field for use in site-specific crop management. Major issues in the implementation of PA include interpreting the huge amount of data collected, understanding the causes of variability, and being able to propose sound strategies for field variability management. To help address these issues, a software infrastructure is proposed. It adopts concepts from software engineering with the aim of providing a basis for the development of information systems for PA based on open platforms, and on data communication and software interoperability standards. The infrastructure has two main components and a reference architecture for the development of distributed service-oriented systems. This paper presents and discusses the infrastructure and its components and presents a prototype application named yield data filtering, which filters yield monitor data and was developed as a proof of concept.

Keywords: Information systems; Web services; XML; Precision agriculture; Infrastructure

1. Introduction

Precision agriculture (PA) is a relatively new management concept introduced in the mid-1980s, which besides attracting considerable interest, started a revolution in resource management (Robert, 1999). Since then, it has been given various names (site-specific management, site-specific farming, precision farming, and precision agriculture) but a common underlying characteristic is its data-driven and technology-intensive nature. Although the use of PA techniques and equipment is increasing, the rate of adoption has slowed down compared to the mid- and late 1990s. There is evidence that the further expansion of PA is being delayed for a number of reasons and, according to Griffin et al. (2004), many of these are educational in nature, or are related to difficulties in dealing with the technology. The authors cite "time to learn equipment and software, lack of electronic skills, lack of training for producers and industry, linking data collection and decision making, lack of technical assistance, lack of local experts, working with data of differing formats, yield data analysis for limiting factors of production, difficulty in maintaining data quality, basic research on yield and soil relationships, and need for a PA equipment, techniques, software do's and don'ts or pro's and con's" (Griffin et al., 2004, p. 16). A survey of farmers in Denmark showed that, although they are generally optimistic about PA, a major problem has been the difficulty in verifying the economic and environmental gains (Pedersen et al., 2003).

* Corresponding author. Tel.: +55 11 3091 5366; fax: +55 11 3091 5294.
E-mail addresses: antonio.saraiva@poli.usp.br, amsaraiv@usp.br (A.M. Saraiva).

0168-1699/$ – see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.compag.2006.12.010

Among the barriers that must be surmounted before PA technologies can be widely and rapidly implemented is data overflow for farm management (Zhang et al., 2002). A large amount of data can be electronically acquired in the field. It is also possible to practice variable rate application (VRA) of lime, fertilizers, herbicides, and other agrochemicals. Although there is much scope for improvement in the existing electronic technology for data acquisition and for VRA, these two tasks do not seem to be the main bottleneck at the moment. Instead, the main problems appear to lie between them, in the interpretation of the huge amount of data collected, understanding the causes of variability, and being able to propose sound strategies for field variability management to be used on VRA. That issue has been interestingly presented by Griffin et al. (2004) as the dichotomy between "information-intensive" and "embodied-knowledge" technology. The first allows intensive data acquisition, and leaves to the data owner the task of dealing with it; while the second embeds a lot of the data and knowledge about how to deal with it in technological packages, so as to free the user from having to deal with poorly understood techniques and a lack of knowledge.

The most important requirements for PA include the following (Saraiva et al., 1997, 1998; Lütticken, 2000; Fountas et al., 2002; Sørensen et al., 2002; Backes et al., 2003; Korduan, 2003; Pedersen et al., 2003; Adrian et al., 2005):

- Management and decision support systems should be designed to meet the specific needs of the farmers.
- Systems should have a simple user-interface that allows customization to different user profiles. A friendly user-interface is especially important for those users who are inexperienced with software.
- Automated and simple-to-use methods for data processing are necessary. Systems should allow inclusion and programming of new automated methods according to user-defined rules.
- The user must also be allowed complete control, whenever desired, having access to the parameters for processing and analysis functions. Expert users may wish to control and try new solutions.
- The introduction of expert knowledge (for instance, rule-based knowledge) must be possible. This may offer the opportunity to fine-tune the systems to local conditions, and to include the user's expertise, practices, and preferences (such as risk profiling, for instance).
- More integrated and better standardized computer systems are needed. This might reduce the technical investment, the learning curve, and the need for technical support.
- Support for easy and seamless integration and interoperability with other software packages (including simulation packages), other data sources (such as meteorological data, market data), locally or remotely via the Internet, using open data standards, interfaces, and protocols. This is especially important to accommodate legacy systems and distributed systems.
- Scalability, to serve different needs.
- Support for meta-data to allow data interchange between applications.
- Low cost.

It is unlikely that a single proprietary system will ever meet all of these requirements because of their complexity and comprehensiveness. That is why an open software platform is a more appropriate solution to the problem (Saraiva et al., 1998; Lütticken, 2000; Murakami et al., 2002). Another important point that favors an open, well structured, component-based solution is that the practice of precision agriculture still has many uncertainties that are the subject of ongoing research. As a consequence, new methods and new processing techniques may have to be incorporated into information systems in the near future, as they become available and experimentally proven. Support for this must be anticipated during the system's development by means of sound software engineering techniques and concepts.

Based on the above considerations, we decided to work on the development of information systems for PA, but from a more conceptual level. Instead of developing simply "another" information system (IS), the purpose was to develop the foundations upon which a class of more flexible, comprehensive, interoperable systems that might meet those cited requirements could be developed under an open software concept. The software engineering paradigm of object orientation and a component-based approach were adopted, as they offer concepts, techniques, and tools that contribute to the achievement of the requirements listed above. A first set of models – use-case models and class models – was created that set the basis for understanding the domain of precision agriculture ISs. These models were named MOSAICo, as they constituted a set of components that together modeled the PA domain (Saraiva et al., 1997, 1998). They were high-level abstract models, technology-independent and free of implementation details so as to guide future detailing towards implementation.

In the second step, a software infrastructure was developed to offer further guidance to IS implementation. This infrastructure was initially based on a set of software frameworks, one of them being the xMOSAICo, an extension of the original model, and on the use of eXtensible Markup Language (XML), which is becoming a key option for achieving systems interoperability (Murakami et al., 2002). A second reason for the adoption of this new technology and the underlying architectural concept of service orientation (SOA—Service-Oriented Architecture; Korotkiy, 2005) was because they provide a strong support to interoperability and to distributed systems, not only theoretical but also practical, taking into account that the software industry is supporting, developing and adopting them.

The objective of this paper is to advance one step further. We present and discuss an infrastructure proposed for the development of distributed service-oriented systems for PA, which is composed of: a reference architecture; a standard language for data exchange between systems, named PAML – Precision Agriculture Markup Language; a service bus, AgriBUS – a message-oriented middleware for connection of Web Services. We also present a prototype application, a service for yield map data correction—named Yield Data Filtering, which was developed as a proof of concept.

2. The infrastructure

The proposed infrastructure has a reference architecture, a standard language for data exchange between systems and services based on XML, and a service bus as a message-oriented middleware for connection of Web Services. The following sections show the reference architecture and the components.

2.1. Reference architecture for precision agriculture information systems

The reference architecture for PA information systems is based on MOSAICo (Modelo de Objetos de Sistemas Abertos de Informação de Campo) which is an object model for PA information systems (a reference model) in the service-oriented architectural style. The reference model serves as a guide in terms of systems functionalities within the domain of PA and the architectural style helps in obtaining architectural properties such as extensibility, changeability, and reusability. An overview of the reference architecture is presented in Fig. 1.

Clients such as Web browsers, WAP-phones, or pagers access the applications (App) through the Portal that centralizes and provides simple and standard interfaces to the users. The applications interact with the Service Bus, which is responsible for invoking the services that are physically distributed across the network. The bus must guarantee the delivery of the requisitions and, if necessary, may apply transformations to the data before communicating with the services. The Service Bus uses a Repository to store the results of those services that demand great processing effort and are, thus, asynchronously invoked. Both the Agricultural Services and the Geospatial Services are Web Services hosted by application servers distributed throughout the Internet. The Agricultural Services perform complete tasks of a business process; for instance, plant growth simulation, data capture (meteorological data, soil data, etc.), and yield monitor data filtering, and are built upon the paradigm of service-oriented computing (Huhns and Singh, 2005). The Geospatial Services are spatial data manipulation services typical of a GIS (Geographic Information System) and are based on the standardized services of the Open GIS Consortium, OGC, Web Feature Service (WFS) and Web Map Service (WMS) (OGC, 2006). The communication of both Agricultural and Geospatial Services is based on the exchange of XML documents which are created and validated by the schema PAML (Precision Agriculture Markup Language), which is presented later in the text.

2.1.1. Web services

The reference architecture proposed for PA information systems basically uses the service oriented architectural style, SOA – Service-Oriented Architecture (Korotkiy, 2005). SOA is frequently characterized as a style that supports loose coupling, business alignment and Web-based services, which permits extensibility and interoperability independent of the technology. SOA architecture allows obtaining a loose coupling between its processing components (service consumer and service provider), because, it uses simple generic and application-independent connectors, and messages defined by an extensible XML schema (Fallside and Walmsley, 2004) for exchanging information with other connectors. Since they are generic, the specific semantics of the application must be expressed in descriptive messages that communicate the problem description from a consumer to a service provider. These messages specify what must be solved but not how it has to be solved, since the service provider is the one who is capable of solving the problem.

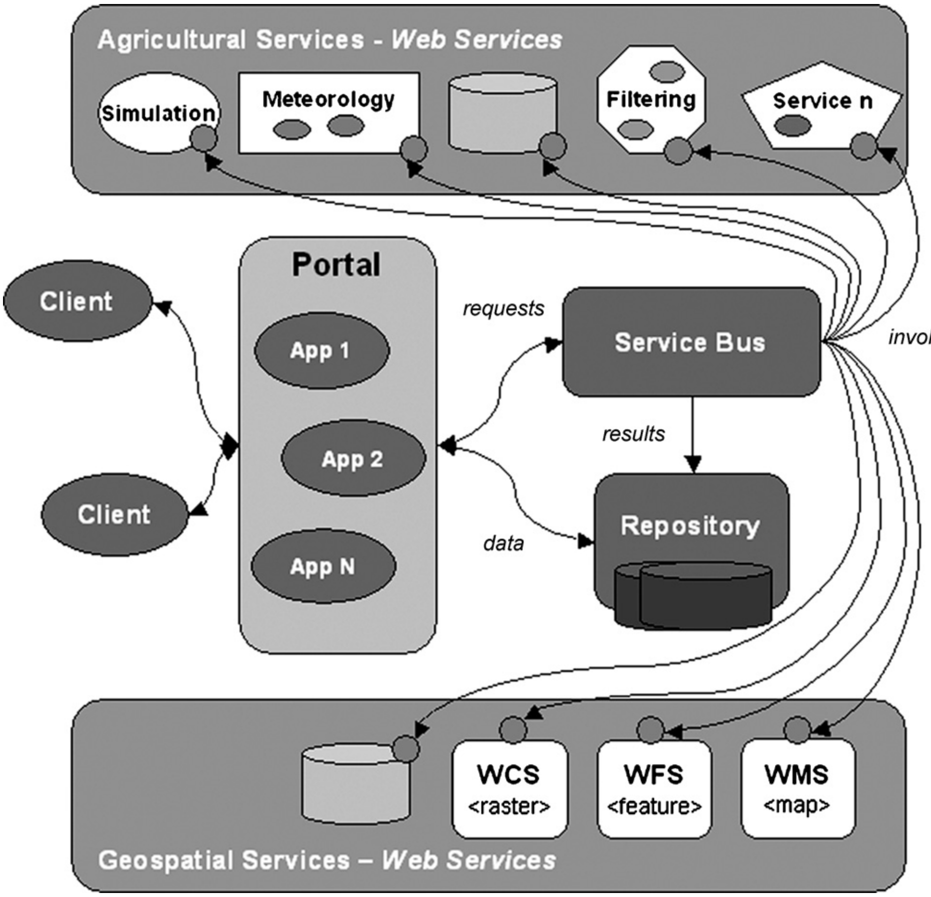

Fig. 1. Overview of the reference architecture. A bus service receives requisitions from the portal applications and invokes the appropriate services, Agricultural or Geospatial. When processing is finished it stores the results in the repository and notifies the client.

Web Services (Stal, 2002; Booth et al., 2004) are the basic components of distributed service-oriented systems. The World Wide Web Consortium, W3C, defines Web Services as a software system designed to support machine-to-machine interaction over the Internet (W3C, 2006). The main difference between Web Services and traditional approaches, such as the distributed objects technologies from the Object Management Group (OMG), Common Object Request Broker Architecture (CORBA) (Vinoski, 1997) or the Distributed Component Object Model (DCOM) from Microsoft (Horstmann and Kirtland, 1997), lies in the aspect of loose coupling of the architecture. Instead of building applications that result in collections of objects or components that are firmly integrated, that are well-known and understood in development time, the service approach is much more dynamic and is able to find, retrieve, and invoke a distributed service dynamically. Another key difference is that with Web Services the industry is solving problems using technologies and specifications that are being developed in an open way, via partnerships and consortia such as the W3C and the Organization for the Advancement for Structured Information Standards (OASIS), and using standards and technologies that are the basis of the Internet.

2.2. A standard language for data exchange (PAML)

In order to allow services interoperability in a business domain, a common vocabulary is needed for automatic and independent information interchange. A standard language for data exchange in the domain of precision agriculture was proposed, PAML, Precision Agriculture Markup Language. For its specification open standards were used, recommended by organizations such as W3C (World Wide Web Consortium) and OGC (Open GIS Consortium). Adopting standards is fundamental to guarantee services quality, accessibility, interoperability, extensibility, and util-

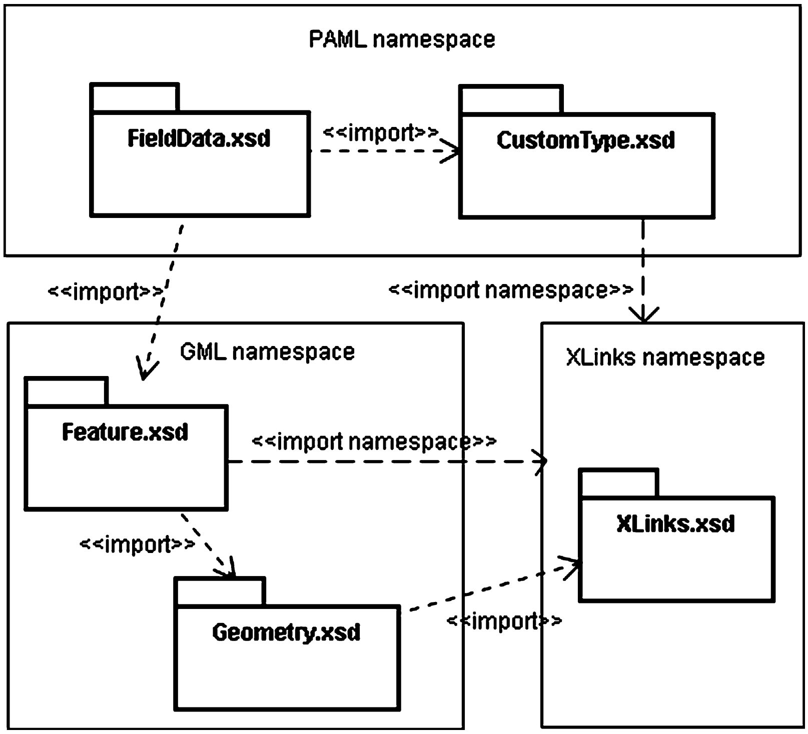

Fig. 2. Abstract model of PAML.

ity. The PAML vocabulary was defined based on the model MOSAICo (Saraiva et al., 1998) and on AGROVOC, a controlled terminology vocabulary for agriculture and related domains kept by the United Nations Food and Agriculture Organization, FAO (Agrovoc, 2006).

PAML was implemented creating an extensible XML schema (Fallside and Walmsley, 2004) capable of representing geographic objects from the real world. It extends the Geography Markup Language (GML; Cox et al., 2002). GML has two object models: *Geometry* and *Feature*. The *Geometry* model defines geometries such as *Point*, *LineString*, *Polygon* and *LinearRing*, and also sets of geometries, both homogeneous and heterogeneous. Geographic properties of real-world entities must be represented by classes of this model. The *Feature* model is used to represent geographic phenomena which are phenomena of the real world and have a position relative to the Earth. A *Feature* may or may not have geographic properties that will be expressed by a relationship with a *Geometry*.

The PAML namespace is composed of the vocabularies defined in two XML schemas FieldData.xsd and CustomType.xsd. The first contains the Field data model from MOSAICo and the second has vocabularies and structures for validation, such as countries list, formatting of specific data such as date, etc. The GML namespace contains the objects models *Geometry* and *Feature*. The Xlinks namespace contains the Xlink.xsd schema. It defines the XML Linking Language (XLink) which allows creating links to resources in XML documents. In other words, it allows the description of hyperlinks in XML documents in the same way as is done in Hyper Text Markup Language, HTML, documents. Fig. 2 shows the abstract model of PAML, which extends the geometric and conventional properties of GML.

Fig. 3 shows the PAML object model. All the classes that extend from AbstractFeature are features of PAML. That inheritance relationship guarantees compatibility with GML. Those classes are part of the module Field Data of the model MOSAICo. The class FieldDataModel extends AbstractFeatureCollection, which has relationships with the main features of PAML. Those relationships are defined on the XML schema FieldData.xsd.

2.3. The service bus—AgriBUS

AgriBUS is a Java implementation of an Enterprise Service Bus (ESB) based on the ESB proposed by Sodhi (2005). ESB is a new middleware technology that provides the characteristics required by SOA. It provides an environment for

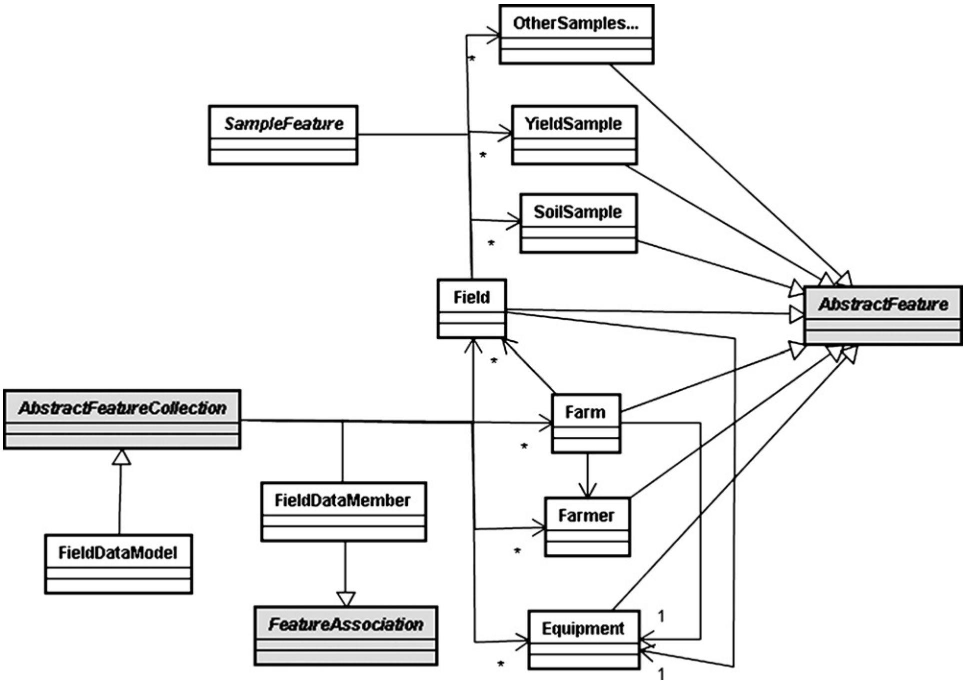

Fig. 3. Object model of PAML.

hosting Web Services allowing the connection and the exposition of services over transport protocols based on Internet standards, such as HTTP (Hyper Text Transfer Protocol) and FTP (File Transfer Protocol). ESB provides characteristics that are essential for the implementation of services such as management, transformation, and validation of messages (Pasley, 2005).

AgriBUS (Fig. 4) supports the more common functional requirements of an ESB:

- *Routing*: to provide a routing mechanism that is flexible and efficient.
- *Transformation*: based on the requester and on the target, to be capable of applying transformation to the data so that the target can understand it.
- *Transport multiprotocols*: to be extensible enough to support multiple message protocols according to the needs of the user.
- *Security*: to provide mechanisms for authentication and authorization for accessing different services.

AgriBUS is formed by the Business Applications, Technological Platform Applications, and the Service Bus. The main components of the Service Bus – receivers, core, and dispatchers – are shown in Fig. 5.

Receivers: two types of interfaces are exposed to allow client applications to send messages to AgriBUS: one servlet to receive HTTP requisitions and one destination Java Message Service (JMS; Roman et al., 2002).

Core: this part of AgriBUS is responsible for routing, transformation, and security application. It is composed of a Message-Driven Bean (MDB; Roman et al., 2002), which receives the requisitions and, based on the message context, applies the necessary transformations, routing, and security. The information about transformations, routing, and security are specified on the message's properties.

Dispatchers: consists of manipulators of the different types of transport mechanisms of AgriBUS such as email, HTTP, and message services.

AgriBUS supports the communication via XML (PAML) documents, a fundamental requirement for integration, and permits applications to communicate synchronously through requisition and answer using HTTP, and asynchronously through message exchange using JMS. This combination of service-oriented architecture, message-oriented architecture and XML makes AgriBUS a robust connectivity provider for services based on various types of software such as legacy applications, databases, Enterprise JavaBeans (EJB) components (Roman et al., 2002), CORBA, and Web

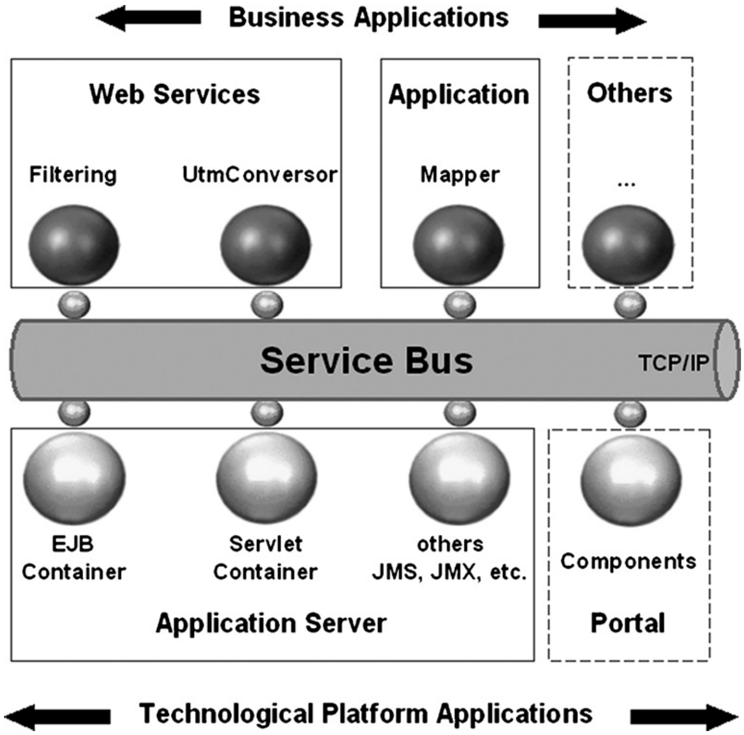

Fig. 4. High-level architecture of AgriBUS.

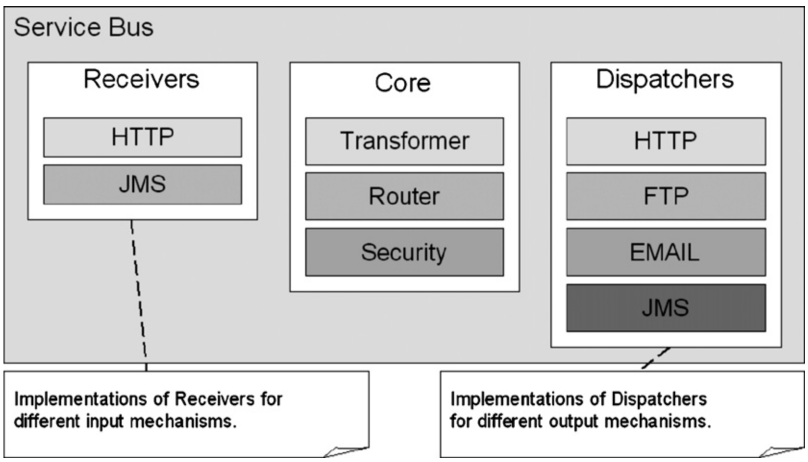

Fig. 5. Components of the service bus of AgriBUS.

Services, making them available for reuse and providing the users with service qualities such as reliability, fault-tolerance, and security; characteristics that are fundamental to supporting transactional activities in the electronic agribusiness.

3. Discussion: prototype yield data filtering

In order to evaluate the proposed infrastructure, to serve as a proof-of-concept, to guide its evolution, and to evaluate and to define technical solutions, a prototype was implemented. This prototype is an application named Yield Data

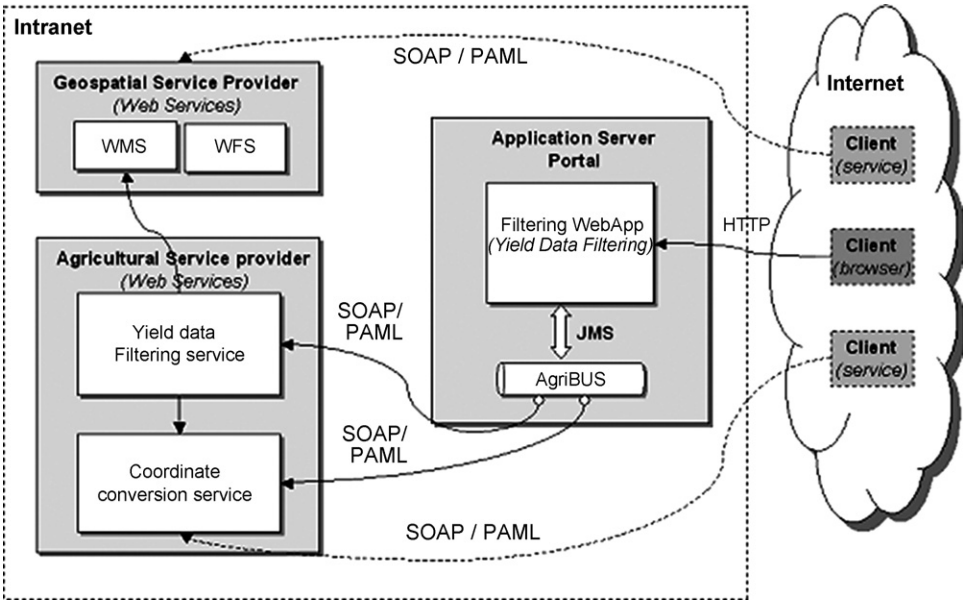

Fig. 6. Overview of the application Yield Data Filtering.

Filtering, which filters yield monitor data based on the algorithm for yield map error removal described by Molin and Menegatti (2002). It specializes in the identification, characterization, and removal of error data points from yield monitor data files.

The architectural style layered-client/server (Garlan and Shaw, 1993) was used in the architectural project of this prototype. Layered systems reduce coupling through multiple layers by hiding the internal layers from the others, except for the adjacent layers. This also improves software reusability and provides high flexibility for changes. Each layer, for instance, can be located in different machines; changes on the data presentation do not alter the lower layers and changes on the physical localization of the services are resolved by the automatic generation of new connectors, from their WSDL (Web Services Definition Language) descriptors. The disadvantage is that they add an overhead and latency to data processing, reducing performance. This disadvantage is compensated for by the improved distributed processing capacity and by the scalability provided by the adopted technologies.

The Presentation layer is responsible for data presentation, interaction, and reception of client requisitions and for forwarding them to the adjacent layer.

The Middleware layer is responsible for decoupling the business layer to allow extension and reuse of the business logic.

The Business layer is responsible for processing the business logic. This layer is composed of services (Web Services) that receive requisitions through open standard protocols and that communicate through a standard language. The objective is that they should be platform-independent, scalable, extensible, and interoperable.

Fig. 6 gives an overview of the application Yield Data Filtering within the proposed architecture. At the presentation layer an application, Filtering WebApp, was implemented using the Struts framework, which implements the design pattern Model-View-Controller (Struts, 2004). Filtering WebApp is responsible for receiving the requisitions of the clients (Client) through a Web-based graphic interface and for forwarding them to the adjacent layer. That interface involves uploading two text files: one with the field border data and the other with the yield data from the same field. A few parameters are also required, which refer to the order of the data inside the file, to allow mapping and conversion of the data from the text format to PAML format, thus generating an XML file. The output is a text file in the same format as that of the input file, without the error data that were removed. The file can be retrieved via download.

The Middleware layer is logically represented by the service bus (AgriBUS). Filtering WebApp sends requisitions to the service bus (AgriBUS), where the clients responsible for accessing their respective service providers Yield Data Filtering Service and Coordinate Conversion Service are connected.

AgriBUS receives JMS messages and puts them into a queue; they are consumed by a controller that verifies the target defined on the message header, converts them to SOAP messages, and sends them for processing. The services at the Business layer receive the SOAP messages with PAML files attached, process them and return an answer. Upon receiving that answer, the controller puts it in a queue and informs the client by email that the processing has finished. The client then downloads the file with the filtered data. The main benefit of this layer is the transparent use of AgriBUS as middleware in the asynchronous communication, with guaranteed message delivery and the capacity to manage and monitor the services that are being executed.

The Business layer consists of distributed services (Web Services) that implement the processors of the business logic for Agricultural Services and for Geospatial Services. The main service developed for the prototype was the Yield Data Filtering Service that implements the algorithm for eliminating yield data points with errors, as developed by Molin and Menegatti (2002). Another service was implemented to allow conversion between different coordinate systems (from geographic to UTM and vice versa), the Coordinate Conversion Service.

The prototype, though simple, has demonstrated the feasibility of implementing information systems for PA that are interoperable, low-cost, Web-based, and which have a high evolution capacity. It uses an infrastructure that is based on open standards and open software platforms. The demonstration of how to build it was operational; constructing the business rules as distributed services (Agricultural and Geospatial) interconnected via a service bus and accessed through user interfaces based on Web standards available in a portal.

The prototype allows users to access the portal on the Web, send their yield monitor data files for error filtering, and receive the files back with the filtered data. Although the main purpose of this first Web application was only the filtering algorithm, other services had to be developed to allow the completion of that task. They are a coordinate conversion service and a map generator service.

The filtering algorithm is an implementation in Java language of the methodology developed by Molin and Menegatti (2002) and Menegatti and Molin (2003) for yield monitor raw data filtering. It consists of a procedure with eight steps to eliminate errors usually present in yield maps. Originally the following eight steps were executed semi-automatically by the authors with the help of an electronic spreadsheet: prepare data for filtering; remove coarse positioning errors; remove points with zero or absent productivity; remove points with partial platform swath; remove data with zero or absent moisture content; remove points with zero distance; remove points recorded during filling time; remove points with discrepant yield values.

These steps were implemented as a service that is accessed through a Web application that provides a graphic user interface (Fig. 7). The service was tested using the same yield data used by Menegatti and Molin (2003) which came from various yield monitors available in Brazil (RDS® Ceres 2, RDS® Pro Series 8000, AFS®, FieldStar®, GreenStar® and New Holland®) working on fields with size ranging from 12.6 to 42.2 ha.

We applied the web filtering on maize samples collected by the RDS Ceres 2 in an area of 22.0 ha with 12,022 points and then we compared the results to those obtained by Menegatti and Molin (2003) using the semi-automatic procedure. In the semi-automatic procedure 1674 points of the samples were removed, while with the web filtering service 1690 points were removed. Although the objective here is not to compare the accuracy of the implementation but to validate the concepts and the infrastructure, the difference between them was only 16 points (less than 1%) and can be attributed to errors during the semi-automatic procedure, which is more prone to errors. Figs. 8 and 9 show the corresponding maps before and after the filtering process using the filtering service.

The coordinate conversion service is used by the filtering service and consists of a Java implementation of a conversion between geographic and UTM coordinates and vice-versa.

The map generation service is used for generating maps on the web from the filtered data. It uses an open source Java implementation of the OpenGIS Consortium standard services for web Geospatial Services, Web Map Service and Web Feature Service (Percivall, 2003), which is called Deegree (Deegree, 2006).

Other services can be easily added to the system as they become available. They will benefit from the existence of the infrastructure already developed, which will guide the development of new functionalities in order to maintain the important characteristics obtained (interoperability, flexibility, distribution). These new services do not have to be developed by the same institutions or groups, nor even need to be hosted in the same place. They can be developed and hosted anywhere, provided that they are published on a registry and are available for use.

If services are developed by many collaborators we might eventually have most of the processing needed for PA available on the Web. This will follow the most recent trend of Web computing, according to which even text editors will be executed on the Web. A more conservative possibility would be the use of the infrastructure for making available

Fig. 7. Graphical user interface of the web application that uses the Filtering service.

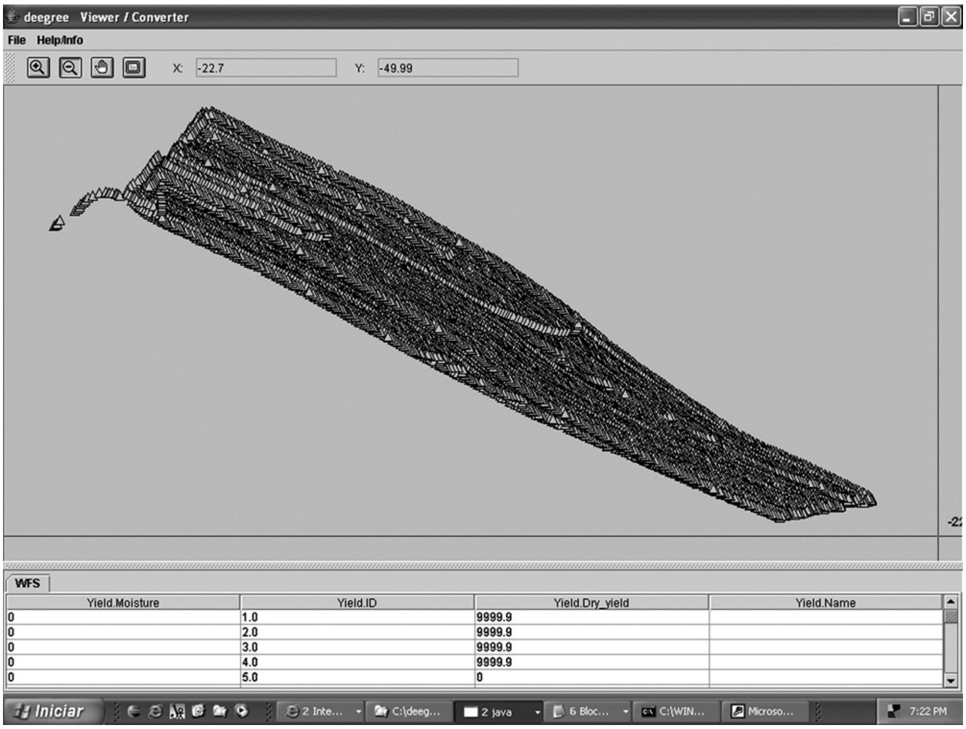

Fig. 8. Map of the raw yield data obtained with the RDS® Ceres 2 yield monitor.

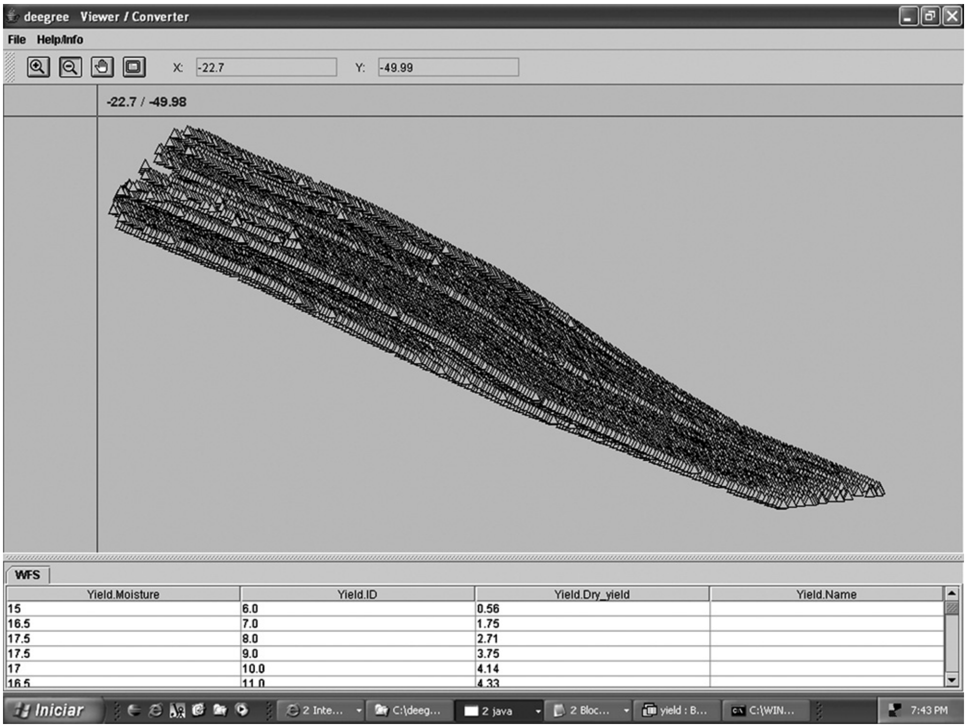

Fig. 9. Map obtained after filtering the raw yield data with the Filtering service according to the algorithm developed by Menegatti and Molin (2003).

specialized algorithms developed by researchers. These algorithms are not usually incorporated into commercial software products, and publishing them as Web Services might be of great value. This does not necessarily mean free access; although that may be desirable, access can be restricted and may require authentication or even payment. It is not difficult to foresee the need for integration into PA information systems, of data, information and processing about the weather, prices (of agricultural produce, inputs, machines, etc.), legislation, and expert technical recommendations that would greatly improve the value of the systems. That new category of system would then get closer to being a real decision-making tool, contributing to more comprehensive, farm-level systems. Many of those services, such as stock prices and weather data, are already required on a real-time basis but are not yet seamlessly integrated into other information systems.

4. Conclusion

The infrastructure developed has proved to be a convenient solution for the development of individual software modules that can be published as services and as web applications. A yield monitor data-filtering algorithm was implemented as a web application and allowed us to validate the infrastructure and the technological solutions adopted. Other algorithms can be used on the development of new services. This will increase the visibility and availability of new ideas and developments originated from research to a wider audience, since they need not to wait to be incorporated into commercial systems. At the same time they can maintain a high degree of compatibility if they are built upon the same concepts and infrastructure.

Acknowledgments

We gratefully acknowledge the contribution of J.P. Molin and L.A.A. Menegatti, from Escola Superior de Agricultura Luiz de Queiroz, Universidade de São Paulo, Brazil, whose algorithm for data filtering was used as a case study, and for the discussions about PA and information technology.

References

Adrian, A.M., Norwood, S.H., Mask, P.L., 2005. Producers' perceptions and attitudes toward precision agriculture technologies. Comp. Elect. Agric. 48, 256–271.

Agrovoc, 2006. Agrovoc Thesaurus. Food and Agriculture Organization of the United Nations. Available from http://www.fao.org/aims/ag_intro.htm (accessed 28-01-2006).

Backes, M., Dörschlag, D., Plümer, L., 2003. A metadata profile for precision agriculture based on ISO 19115. In: Stafford, J., Werner, A. (Eds.), Proceedings of the 4th European Conference on Precision Agriculture. Academic Publishers, Wageningen, pp. 41–46.

Booth, D., Haas, H., McCabe, F., Newcomer, E., Champion, M., Ferris, C., Orchard, D., 2004. Web Services Architecture, W3C Working Group. Available from http://www.w3.org/TR/ws-arch (accessed 27-01-2006).

Cox, S., Cuthbert, A., Lake, R., Martell, R., 2002. GML Implementation Specification, version 2.1.2. OpenGIS Implementation Specification. Available from http://www.opengeospatial.org/docs/02-069.pdf (28-01-2006).

Deegree, 2006. Deegree—Free Software for Spatial Data Infrastructures. Available from http://www.deegree.org/ (accessed 10-01-2006).

Fallside, D.C., Walmsley, P., 2004. XML Schema Part 0: Primer Second Edition. W3C Recommendation. Available from http://www.w3.org/TR/xmlschema-0 (accessed 27-01-2006).

Fountas, S., Blackmore, S.B., Petersen, S.M., 2002. A new methodology for decision analysis on precision farming based on user's experience. In: Robert, P.C. (Ed.), Proceedings of the 6th International Conference of Precision Agriculture. ASA/CSSA/SSSA, Minneapolis, MN, USA, pp. 1640–1654.

Garlan, D., Shaw, M., 1993. An introduction to software architecture. In: Ambriola, Tortola (Eds.), Advances in Software Engineering and Knowledge Engineering, vol. II. World Scientific Pub Co., Singapore, pp. 1–39.

Griffin, T.W., Lowenberg-DeBoer, J., Lambert, D.M., Peone, J., Payne, T., Daberkow., S.G., 2004. Adoption, Profitability, and Making Better Use of Precision Farming Data. Staff Paper #04-06. Department of Agricultural Economics, Purdue University.

Horstmann, M., Kirtland, M., 1997. DCOM Architecture. Microsoft Corporation. Available from http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dndcom/html/msdn_dcomarch.asp (accessed 01-02-2006).

Huhns, M., Singh, M.P., 2005. Service-oriented computing: key concepts and principles. IEEE Internet Comp. 9, 75–81.

Korduan, P., 2003. Standardization in data management to increase interoperability of spatial precision agriculture data. In: Stafford, J., Werner, A. (Eds.), Proceedings of the 4th European Conference on Precision Agriculture. Academic Publishers, Wageningen, pp. 323–328.

Korotkiy, M., 2005. Towards an ontology-enabled service oriented architecture. In: Proceedings of the, PhD., Symposium in International Conference on Service Oriented Computing, (ICSOC'05), Amsterdam, The Netherlands.

Lütticken, R.E., 2000. Development of an Internet-based communication and information network to progress the implementation of precision agriculture. In: Proceedings of Fifth International Conference on Precision Agriculture (CD), 2000, Bloomington, MN, USA.

Menegatti, L.A.A., Molin, J.P., 2003. Metodologia para identificação e caracterização de erros em mapas de produtividade. Revista Brasileira de Engenharia Agrícola e Ambiental 7 (2), 367–374.

Molin, J.P., Menegatti, L.A.A., 2002. Methodology for identification, characterization and removal of errors on yield maps. In: ASAE Annual International Meeting /CIGR XVTH World Congress, 2002, Chicago. St. Joseph, ASAE.

Murakami, E., Ribeiro, L.C.M., Saraiva, A.M., Cugnasca, C.E., 2002. An infrastructure for development of information systems for precision agriculture. In: Proceedings of 6th International Conference on Precision Agriculture, Minneapolis, ASA/CSSA/SSSA, pp. 1712–1722.

OGC, 2006. Open Geospatial Consortium. OpenGIS Specifications. Available from http://www.opengeospatial.org/specs/ (accessed 28-01-2006).

Pasley, J., 2005. How BPEL and SOA are changing Web services development. IEEE Internet Comp. 9, 60–67.

Pedersen, S.M., Fountas, S., Blackmore, S., Pedersen, J.L., Pedersen, H.H., 2003. Adoption of precision farming in Denmark. In: Stafford, J., Werner, A. (Eds.), Proceedings of the 4th European Conference on Precision Agriculture. Academic Publishers, Wageningen, pp. 533–538.

Percivall, G., OGC Reference Model, version 0.1.3, 2003. Open Geospatial Consortium. Available from http://www.opengeospatial.org/specs/?page=orm (accessed 28-01-2006).

Robert, P.C., 1999. Precision agriculture: research needs and status in the USA. In: Stafford, J.V. (Ed.), Proceedings of the 2nd European Conference on Precision Agriculture, Part 1. Academic Press, SCI/Sheffield, pp. 19–33.

Roman, E., Ambler, S., Jewell, T., 2002. Mastering Enterprise JavaBeans, 2nd ed. John Wiley and Sons, Wiley Computer Publishing.

Saraiva, A.M., Massola, A.M.A., Paz, S.M., 1997. Object oriented approach to the development of a field information system. In: Annual International Meeting of the American Society of Agricultural Engineers, 1997, Minneapolis. ASAE Paper 97-3015.

Saraiva, A.M., Massola, A.M.A., Cugnasca, C.E., 1998. An object model for field information systems. In: Proceedings of the 4th International Conference on Precision Agriculture, St. Paul, MN, USA. ASA/CSSA/SSSA, pp. 1355–1366.

Sodhi, B., 2005. Implement a customizable ESB with Java. JavaWorld. Available from http://www.javaworld.com/javaworld/jw-08-2005/jw-0808-esb.html (accessed 01-02-2006).

Sørensen, C.G., Fountas, S., Blackmore, S., Pedersen, H.H., 2002. Information sources and decision making on precision farming. In: Robert, P.C. (Ed.), Proceedings of the 6th International Conference of Precision Agriculture. Minneapolis, MN, USA. ASA/CSSA/SSSA, pp. 1683–1695.

Stal, M., 2002. Web Services: Beyond Component-based Computing Communications of the ACM 45, pp. 71–76.

Struts, 2004. Struts Framework. The Apache Software Foundation. Available from http://struts.apache.org (accessed 28-01-2006).

Vinoski, S., 1997. CORBA: integrating diverse applications within distributed heterogeneous environments. IEEE Commun. Magazine 14, 46–55.

W3C, 2006. The World Wide Web Consortium. Web Services Activity Statement. Available from http://www.w3.org/2002/ws/Activity (accessed 24-05-2006).

Zhang, N., Wang, M., Wang, N., 2002. Precision agriculture—a worldwide overview. Comp. Electr. Agric. 36, 113–132.