DATA INFRASTRUCTURES IN AGRICULTURE
Attempts at Interoperability

Daniel Martini and Mario Schmitz
Association for Technology and Structures in Agriculture (KTBL), Bartningstraße 49, Darmstadt, Germany

Keywords: Agriculture, Interoperability, Service infrastructure, Web oriented architecture.

Abstract: Agriculture presents itself as an interesting conglomerate of different domains. It is an intersection of a bunch of natural sciences like biology, chemistry, geography with business, legal and political issues. As diverse as the disciplines involved in agriculture are the demands on data management and exchange. This poses a special challenge on interoperability of data formats and services. Additional requirements arise from the size and structure of enterprises involved in farming and in provision of farm management information systems. From our work in agricultural data exchange, we present experiences and try to derive how data standards touching this domain should develop to allow for future interoperability.

1 INTRODUCTION

Whereas other industries can produce under mostly controlled conditions in closed systems like factories, agriculture is embedded into a natural, economic and political environment with diverse influencing factors, some of them only partly well understood. Farm management today is the art of correct interpretation of a variety of information from different sources describing this environment. Electronic data acquisition and analysis play an increasing role for the farmer. However, data has to be retrieved from a number of places. It has to be handed on to parties imposing different requirements on structure or format than the ones present during original production of the data. Different analysis methods demand for different points of view and aggregation of data. All this leads to major challenges in interoperability and on how to build service infrastructures in agriculture. Current state-of-the-art is far from facilitating data exchange for the farmer. There are diverse standards playing a role in agriculture. A lot of the data items present are not easily mapped from one to another either due to fine semantic differences or inconcise conceptualizations. As the farm management information systems developers cannot judge on these issues themselves either, most decisions are left to the user leading to a lot of human interaction in advance of a data exchange.

In this paper, we present the types of data, standards, requirements on service infrastructures and potential uses of semantic technologies and ontologies in agriculture. It is by no means an exhaustive scientific analysis of the problem. Rather, the paper shows experiences gained during standardization of agroXML, an XML based standard in agriculture in Germany. We will derive the need for action and possible further activities in the area of knowledge acquisition and management in the agricultural domain.

2 TYPES OF DATA AND THEIR SOURCES

2.1 Geographic Data

The field is the central production unit in arable farming. Even the basic geometry of the field has to be considered to achieve optimum performance during production. The decision into which direction to till has a major influence on the economical outcome of work procedures. New methods in farming like e. g. precision farming continue to deliver an ever increasing amount of spatially referenced data. Harvested amounts per small scale area units are determined automatically by sensors in combine harvesters and later on produced into maps. Future measures in the field can be fine-tuned according to results gained from analysing these maps. In some cases, an overlay with further geographic data like e. g. remote sensing data

Martini D. and Schmitz M. (2009).
DATA INFRASTRUCTURES IN AGRICULTURE - Attempts at Interoperability.
In *Proceedings of the International Conference on Knowledge Engineering and Ontology Development*, pages 287-292
DOI: 10.5220/0002308202870292

or aerial imagery is necessary.

A lot of representational aspects of spatial data are standardized by the Open Geospatial Consortium (OGC). The Geography Markup Language (GML, (Cox et al., 2004)) provides basic data types for spatial objects like polygons, points and linestrings. The standard is well designed allowing reuse of parts in other XML based vocabularies using either profiles or imports of necessary schema modules. However, as such, it is a meta-standard. It does not say anything about how to represent certain real world objects. If GML is to be used in a service, developers usually provide their own schema following the feature-property model mandated by the specifications. While this ensures unified modelling procedures, it does not provide for any kind of (automated) interoperability between different OGC conformant web services. Semantic interpretation and agreement on the content delivered is up to the developers of the systems which are to be connected.

2.2 Data on Operating Supplies

Operating supplies used in agriculture have a variety of properties of interest to the farmer. In some of the cases, this information is provided as a printed accompanying sheet. Seeds for example have to have variety information and other quality parameters like germinating capacity determined and printed onto labels accompanying product batches. However, there are also cases, where additional information might be retrieved in digital form. Plant variety offices often have further data on average quality parameters derived from field trials. In Germany for example, this information is available in downloadable comma separated value files.

In pesticide usage, the farmer has to follow application rules. In most cases, properties of pesticides necessary to correctly implement these rules like e. g. the waiting period between application and harvest, have to be published by agencies after tests and an approval procedure. While this information can often also be retrieved via the internet, it is in most cases embedded in a poorly structured form into web pages not ready for automated extraction and reuse in agricultural software systems.

Roughly the same goes for veterinary drugs. Approval and application of pharmaceutics are strongly regulated in Europe. A lot of data is available through European Medicines Agency (EMEA), however documents are published in a variety of more or less structured and standardized formats (Microsoft Word or PDF files amongst them).

All in all, data as such is readily available. Lacking are however machine-readable forms allowing for reuse in decision support systems.

2.3 Laboratory and Animal Data

Results from laboratory analysis present a major part of data in plant production (e. g. soil analysis) and livestock farming. Milk recording and analysis is an area, where mass data acquisition is quite common. Using these data, the breeding value of single animals can be determined by comparing to a large number of individuals in the respective population. The algorithms are complex and calculations are mostly done as a service to farmers by computing centres run by breeding associations. Data exchange for these services is supported by a set of standards: ISO-17532 (ISOagriNet, (Internationals Standards Organisation, 2007)) provides the underlying protocol, whereas the Agricultural Data Element Dictionary (ADED) provides the content. The data dictionary is separated into a part providing data entities for exchange on the national level and into another part providing internationally harmonized entities. There are two different serializations for the format: ADIS/ADED and XML/ADED. The protocol is record oriented and tuned for bulk transfer of large amounts of data. As such, it is not yet ready for easy integration into more web-oriented data networks using standard internet technologies. On-going work however tries to achieve a more friendly serialization in the face of current developments in data exchange.

The problems faced by developers and users of these standards currently mostly lie in the semantics. The data dictionaries consist of a lot of entities and items, and it is often difficult to judge, if the respective use case has already been worked on by another group and if there are already items available, which could be used. Also occuring is the case, that the same item is used with different meanings in different contexts. It is currently not quite clear, how to solve these issues.

2.4 Business Data

Apart from the work in the field or in the stable, the farmer wants to generate an income for himself, his workers and his family. In other words: he is involved into economic activity and conducting business. Standards available in the financial sector are numerous. However, in a lot of cases, they have a very clear and limited scope. The Home Banking Computer Interface (HBCI), the newer FinTS, Interactive Financial Exchange (IFX) or Open Financial Exchange (OFX) for example allow for conducting financial transac-

tions using a Personal Computer and a suited application program. The interface standards are probably of no larger relevance to the knowledge management on a farm, as messages are short lived and are mainly used to change or request state of bank accounts and depots.

Completely different in this regard is the eXtensible Business Reporting Language (XBRL). It is a typical XML application in the sense that it is document oriented and allows linkage between different sections of documents using standard technologies like XLink. The scope is limited to business reports and its use cases are well defined.

On the other hand, there are standards in the business sector, which try to standardize on a syntax for each and every use case one can think of. Within thescope of UN/CEFACT standardization a number of working groups are modelling use cases in all kinds of domains, also in agriculture. The UN/CEFACT XML Naming and Design Rules (Heilig et al., 2006) derive in many ways from best practices recommended by the W3C. It is very difficult to reuse UN/CEFACT schemas together with other XML vocabularies or the other way round. This is mostly due to the fact, that most of the commonly used extension mechanisms (like e. g. inheritance by extension, anyTypes etc.) are forbidden in the specification.

2.5 Supply Chain Data

As soon as primary production on the farm is finished, the goods enter the food chain. There again, another set of standards becomes important, the ones to identify and describe products in a supply chain. The EPC bar codes are probably known to a broader public from the local discounters cashier scanners. However, the organization behind them, GS-1 also produces standards for information systems and services revolving around the product identification like e. g. Electronic Product Code Information Services (EPCIS, (EPCglobal Inc., 2007)). These standards are also limited in scope, allowing for representation of events and basic object data in the supply chain. They are well suited to use cases in traceability of agricultural goods, however difficulties to represent events like distribution of larger amounts into smaller batches (e. g. with bulk materials like cereals or with carving up of animals after slaughter) or putting together ingredients to form another product prevent usage in some areas.

3 PROPOSED SERVICE INFRASTRUCTURE

3.1 Requirements

In our experience, the most important requirement for an information exchange infrastructure for agriculture is simplicity. Most of the companies involved in production of agricultural software are small and medium sized enterprises. They can not afford to spend a lot of resources on implementation of overly-complex protocols and standards. It is a major challenge to work out the essential parts in standards and limit the number of degrees of freedom by providing a very generic and simple set of specifications.

Another aspect e. g. in food traceability is (close to) unlimited scalability. A lot of partners are involved in the food chain from farm to fork. It has to be easy to add further systems and data models to an infrastructure.

Data security is also a major demand from the agricultural community, at least in Europe. Data on agricultural processes are treated as business secrets. This is understandable from the viewpoint that the buyer of agricultural products might use this information to control prices to be paid to the farmer.

3.2 Possible Architecture

During a research project we were able to develop a prototype fulfilling a large part of the above mentioned requirements.

The design of the system is based on a web-oriented architecture (Jacobs and Walsh, 2004). It relies on the usage of technology components standardized by the W3C. Key concepts are globally unique identification by URIs, web linking methods, XML as format for the content and a simple protocol, restricted to a few method invocations - currently the Hypertext Transfer Protocol.

The content and document types delivered are provided by agroXML. It offers the necessary elements and datatypes to be able to represent agricultural issues in XML documents. agroXML is defined by a set of XML Schemas and content lists. They are available at http://www.agroxml.de/schema/ and http://www.agroxml.de/content/ respectively.

Unique identification of resources is provided by URIs. We have used the subset commonly known as URLs, as the mechanism to dereference them is simple and standardized. To link documents in a service, XLink (DeRose et al., 2001) is used.

Putting everything together, a ReSTful web service is built. The term ReST is an acronym for Repre-

sentational State Transfer and has been introduced by Roy Thomas Fielding (Fielding, 2000). It is based on the assumption that with a few simple operations to read and write data and a system changing its state depending on the operations issued, any use cases in communication can be represented. Variations of this concept are a basic thread present in information technology history up to now. The Turing machine (Turing, 1936) already relied on this simple principle. The SQL language common in database systems with its INSERT, SELECT, UPDATE and DELETE operations is built on this pattern. One of the principles in early UNIX system development was "everything is a file", thus allowing for manipulation of devices and compute resources using simple file operations like open, close, create, read and write. Later on, Kilov coined the term Create-Read-Update-Delete-pattern (CRUD, (Kilov, 1990)). The currently widespread Hypertext Transfer Protocol (HTTP, (Fielding et al., 1999)) is built around this assumption as well.

Developing a ReSTful web service involves distributing state and functionalities of a service across a set of network objects. Objects are manipulated using only the small set of basic operations provided by HTTP. This is in contrast to services using an RPC-paradigm, where a single network object offers a large number of method invocations.

For demonstration purposes, a prototype around the following use case has been developed:

- For a group of animals the fattening process is over.
- The farmer puts together a batch for transport to the slaughterhouse.
- Data about the animals is readily available.
- Data about the location, where the batch is built, is readily available.
- The task is to summarize the information about this batch and present it in a machine-readable form on the network.

The modules available in agroXML allowed for building small, self-contained documents able to represent single objects involved as resources on the web.

For pigs, data concerning sex, eartag and events related to the animal like weighing or feeding were laid out in XML instances.

For the prototype, we could rely on unique identification of single animals. Nevertheless, it is also possible to address groups of animals, if there are no unique identifiers available. As a drawback, in the latter case no data concerning a single animal can be given.

It is also possible to represent basic information about the farm in agroXML instances. The objects Farm and Pig become resources on the web by assigning URLs to them. To be able to build a batch, the objects of which it is comprised must be linked in. This is achieved by allowing XLinks from the batch XML instance to point to the single animals contained within and to the farm. The modelling is generic, so that other objects might be batched as well by just linking to them.

In total, the following URL-structure is used for the service:

farm data: http://example.com/farms/*

animal data: http://example.com/animals/*

batch data: http://example.com/charges/*

While in the example, all network objects in the service are available on a single domain, the Xlinks can in principle point anywhere. It is possible to integrate data offered on other servers of external information providers into local applications.

The application shown could also have been built using message-oriented remote procedure calls using e. g. SOAP. For this to work, methods for adding pigs to a batch and for retrieving batch data would have to have been defined. However, as SOAP messages are short-lived and as there is no standardized way to reference objects it is very difficult to add further layers like e. g. a set of RDF (Ressource Description Framework, (Klyne and Carroll, 2004)) statements to relate resources to each other. In contrast, resources in a ReSTful webservice are persistent from a clients point of view. This can be the basis to build e. g. a RDF triple store annotating resources with further metadata or describing relationships between objects. As URLs are the principle of identifying objects in RDF as well as in ReSTful web services, the technologies play together quite well.

A further disadvantage of the message-oriented approach using technologies like SOAP is, that the method calls offered by the server and their parameters have to be known to the client in advance. While this is no problem as long as there is only a limited, strictly controlled set of services available (like e. g. internal to an organisation), it leads to severe difficulties if an unlimited set of applications is to be added. While technologies like UDDI theoretically could provide the necessary object and method publication functionality, in networks with a large diversity of system environments dynamic binding of clients during runtime is currently practically infeasible. In contrast, in a ReSTful service network, a client can navigate through an infinitely large set of resources without requiring detailed knowledge of the complete set.

Tools to implement services like the one shown

are available for a variety of hardware platforms from energy efficient mobile devices up to powerful server machines and for almost every programming environment. Basically, only an implementation of HTTP and an XML Parser is required. The implementation is even possible with reasonable effort in lower level programming languages e. g. as a server module to the Apache webserver in C. Thus the architecture offers the necessary simplicity required to work in environments like the farm, where the IT infrastructure is not as sophisticated as in large enterprises. An additional bonus comes from the fact, that ReSTful services are - in contrast to services using RPC coded in SOAP - cacheable, which allows them to be used across high latency links with proxies and caches in between client and server as often seen in rural areas or developing countries.

4 USAGE OF ONTOLOGIES

4.1 Rule based Systems to Express Legal Obligations

A rule based systems is used as a way to store knowledge and to interpret information. Rules define IF-THEN-terms and form a rule base. An inference engine draws conclusions based on the rules together with other information – the facts. That means, if a fact is given, we can deduce one or various other facts. New facts could be interpreted by the rules in a second cycle. The system has to know when all stop conditions are complied and the result is calculated. Facts are information needed by the system to draw conclusions. These could be data collected from databases, user-filled forms or external resources.

An example of such an engine currently in the works looks upon the use case of a farmer wanting to fill out a form to claim for crop subsidies. A rule system, prepared for this procedure, could analyse all information and help him to complete the form correctly. If he grows a crop that is supported by a special programm, the system is able to interpret his information and point out the possibility to request this support too.

The challenge with regard to creating these rule sets currently are difficulties in transferring information available on the farm – mostly derived as documentation of practical work processes – into a representation according to the conceptual framework given in the form. While there are for example codesets for crops available to facilitate handling by information technology in the administration, they achieve the opposite for the farm management information system, in that they mangle and merge existing, separate concepts like species, usage and receiver of a crop or their superclasses into a single coded item. That makes content mapping a very difficult and cumbersome process often requiring human interaction.

In the future, a sophistcated rule system might go a step further too. It can help to take decisions e. g. what kind of fertilizer a farmer might need for a certain variety under certain weather conditions. Based on the kowledge of an expert, the system knows the best answer. The application chooses all conditions – IF-part of a rule – concerning his case und makes the right conclusion provided all important facts are known and the rule base is large enough.

The rules should – if posssible – be expressed in an almost natural language. The idea behind this is that a domain expert should be able to create rules in a language which a computer understands without knowing programming languages. This makes the system not only user friendly, but also flexible.

In such a system, the algorithms don't depend on the implementation or programmed code. With a rule managing system, a domain expert can easily delete or change old rules and add new rules to the application.

Such a rule based system is comprised of four main components:

1. a inference engine that interpretes a broad set of records.
2. a sufficiently large knowledge base to achieve satisfying results.
3. a system that aggregates information and translates them in a form usable in the inference engine
4. a user interface for creating requests and administering the rules

Such a system can only be established based on networking and an exchange of standarized information, such that different islands of technologies an controlled vocabularies merge together.

5 CONCLUSIONS

All in all, agriculture provides an interesting playground for knowledge engineering methods and technologies. As agriculture has to retrieve information from a variety of sources in different domains, interoperability of standards is a major issue. Currently missing is an approach at extensibility and the documentation and implementation of best practices for and by standards developers satisfying the demands

of data exchange according to the world wide web's paradigms. Clarifications of semantics and methods to handle small differences in semantics on transformation from one standard to another correctly are also required in the future.

ReSTful web services provide the simplicity and scalability required in future exchange of agricultural data. They can be seen as a distributed dataset. A system drawing logical conclusions from this dataset comparable to datamining applications on relational database systems could be built. Using technologies like the web ontology language (OWL, (McGuinness and van Harmelen, 2004)), rulesets could be loaded on-demand by client applications for different purposes. Expert systems providing agricultural extension services might profit from such an approach.

REFERENCES

Cox, S., Daisey, P., Lake, R., Portele, C., and Whiteside, A. (2004). *OpenGIS Geography Markup Language (GML) Implementation Specification*. Open GIS Consortium, Inc.

DeRose, S., Maler, E., and Orchard, D. (2001). *XML Linking Language (XLink) Version 1.0*. World Wide Web Consortium. http://www.w3.org/TR/xlink/.

EPCglobal Inc. (2007). *EPC Information Services (EPCIS) Version 1.0.1 Specification*.

Fielding, R. T. (2000). *Architectural Styles and the Design of Network-based Software Architectures*. PhD thesis, University of California, Irvine.

Fielding, R. T., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. (1999). *RFC2616: Hypertext Transfer Protocol – HTTP 1.1*.

Heilig, P., Stuhec, G., Pemberton, M., and Minakawa, G. (2006). *XML Naming and Design Rules, Version 2.0 of 17 February 2006*. UN/CEFACT.

Internationals Standards Organisation (2007). *Stationary equipment for agriculture – Data communications network for livestock farming*.

Jacobs, I. and Walsh, N. (2004). *Architecture of the World Wide Web, Volume One*. World Wide Web Consortium. http://www.w3.org/TR/webarch/.

Kilov, H. (1990). From semantic to object-oriented data modeling. In *Proceedings of the First International Conference on Systems Integration*.

Klyne, G. and Carroll, J. J. (2004). *Resource Description Framework (RDF): Concepts and Abstract Syntax*. World Wide Web Consortium. http://www.w3.org/TR/rdf-concepts/.

McGuinness, D. L. and van Harmelen, F. (2004). *OWL Web Ontology Language: Overview*. World Wide Web Consortium. http://www.w3.org/TR/owl-features/.

Turing, A. M. (1936). On computable numbers, with an application to the entscheidungsproblem. In *Proceedings of the London Mathematical Society*, volume 2 42.