Project no.	**212117**
Project acronym:	**FUTUREFARM**
Project title:	**Integration of Farm Management Information Systems to support real-time management decisions and compliance of management standards**
Instrument:	**Collaborative project**
Start date of project:	**1st January 2008**
Duration:	**36 months**
Thematic Priority:	**THEME 2 FOOD, AGRICULTURE AND FISHERIES, AND BIOTECHNOLOGY**

Deliverable 4.1.1.
Machine Readable Encoding for Definitions of Agricultural Crop Production and Farm Management Standards

Due date of deliverable: **28/02/2009**
Actual submission date: **17/03/2009**

Work package 4: **"Knowledge Management in the FMIS of Tomorrow"**

Organisation name of lead beneficiary for this deliverable:
Rostock University (UR) Revision**: Final**

Authors: **Edward Nash (UR), Raimo Nikkilä (TKK), Liisa Pesonen (MTT), Kai Oetzel (Agrocom), Walter Mayer (Progis), Ilkka Seilonen (TKK), Jere Kaivosoja (MTT), Ralf Bill (UR), Spyros Fountas (CRTH), Claus Sørensen (AU)**

Project co-funded by the European Commission within the Seven Framework Programme (2007-2013)

Dissemination Level		
PU	Public	X
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Summary

This document describes the initial draft of a machine-readable encoding format for the definition of agricultural crop production and farm management standards[1]. These standards may be:

- Regulations defining the legal minimum requirements which must be met by farmers (e.g. fertiliser regulations).
- Legal requirements defining aspects of good agricultural practice and which must be met in order to receive state support (e.g. cross-compliance).
- Legal requirements which must be met in order to produce a particular category or class of product (e.g. organic farming regulations)
- Private standards which farmers may choose to comply with for marketing, financial or other reasons (e.g. GlobalGap).

We assert that all these categories of 'standards' may be modelled and represented in a unified manner as presented here.

The main focus of this document is in documenting the process and design decisions made in producing the encoding format. This includes a use-case analysis of how we expect the encoding format to be used and a statement of the design requirements against which the format was evaluated. Additionally, this document serves as a document of the data model which is implemented by the encoding format.

Version history

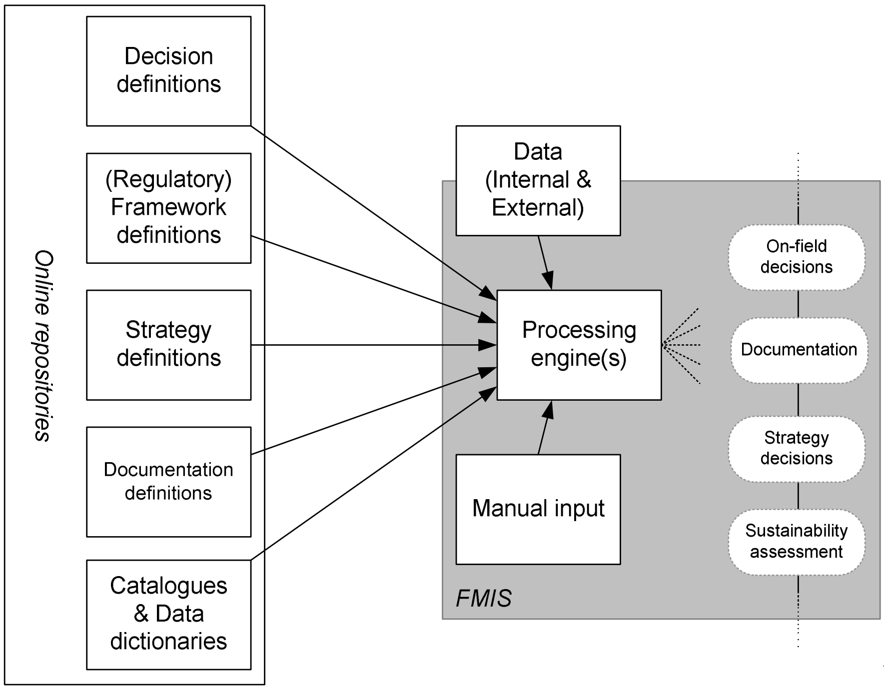
Version	Date	Comment	Last Author
0.1	2009-02-24	Incomplete first draft	Nash
0.11	2009-03-02	Added text on rule encoding and ontologies	Nikkilä
0.12	2009-03-06	Complete first draft without data model	Nash
0.2	2009-03-17	Added data model and changes based on comments received	Nash
Final	2009-07-02	Final cosmetic changes	Nash

[1] The planned title of this deliverable was "Machine-Readable Encoding of Regulatory Framework Requirements" – this has been altered in light of the work carried out.

Table of contents

1. Introduction

This document presents the design decisions involved in producing a machine-readable encoding of agricultural crop production and farm management standards. Fundamental to this is the question, “for what purpose will this encoding be used?” This can be answered with reference to Figure 1, taken from the FutureFarm project proposal: it is expected that a future Farm Management Information System (FMIS) will be able to support on-farm decision making while respecting any and all standards and regulations which are relevant for the farm. Such standards are analysed in Vatsanidou et al (2009). Due to the possible range of these, varying by geographic location, farm type, farmer preference and through time, it is unrealistic to hard-code such standards directly in the FMIS. We therefore propose a Service-Oriented Architecture (SOA) by which machine-readable definitions of these standards may be automatically accessed by the FMIS via the Internet. The FMIS must then be able to interpret these standards together with data held in internal and external databases, or entered manually, in order to guide the farmer in standards-compliant agriculture. In order for this to function, there must exist a standard transfer format with which the knowledge of the standard can be transferred from the server (repository) to the client (FMIS). The development of this transfer format is the focus of this document.



b

Figure 1. Proposal for a distributed FMIS supporting standards-compliant decision-making

Note that the proposed system is intended primarily to support the farm’s internal decision-making processes and the production of documentation required to demonstrate compliance to standards. It is not intended that the system will automate compliance checking, as is currently performed by audits through inspection and certification bodies, although elements of the system could in future be used for this purpose. This restriction has a number of reasons:

- External audits and inspections are an important mechanism for helping prevent fraud.
- The security infrastructure required to securely collect and manage the compliance data would add a large overhead to the system.

- Standards contain requirements which cannot be automatically assessed as they require contextual information which cannot be automatically/digitally collected and/or processed.
- The concept of '*mechanical jurisprudence*' is rejected by experts in the field of law and artificial intelligence as unrealistic – "law is more 'rule-guided' than 'rule-governed' (Gardner, 1987) and interpretation and legal argumentation plays a major role in assessing compliance to laws (Boer et al, 2007). Particularly for the legal regulations a purely automated assessment of compliance would therefore not be tenable as computable semantic interoperability is difficult to obtain in this area.

Particularly the last two of these points pose the question as to whether producing a rule-based system as proposed is a helpful contribution. We believe it is as even where the compliance cannot be automatically assessed, such a system may clearly present the requirements to the farmer, allowing a better-informed decision to be made. We therefore propose that rule assessments which may (straightforwardly) be automated should be automated, and those which may not be (straightforwardly) automated will be presented to the farmer for a manual assessment. Furthermore, for the task at hand in this deliverable, that of developing a machine-readable encoding of agricultural standards, we are not concerned with how these standards will be interpreted, and as Boer et al (2007) state, "one *can* define legal concepts this way, but such definitions are only hypotheses or theories which will not be blindly or 'mechanically' followed, using deduction, when one tries to apply these concepts to decide legal issues in concrete cases". For the simple expression and transfer of the agricultural standards, a rule-based system is therefore sufficient – the fact that some of these rules may not be open to automated interpretation and compliance assessment must however be accepted as a limiting factor when developing an FMIS to interpret them.

2. Structure of Agricultural Standards and Regulations

Standards and regulations may be published in many forms (e.g. paper/electronic, prose/checklist) and by many organisations. Fundamentally, however, all standards and regulations have the same underlying structure (Figure 2):

- Metadata including some or all of:
  - The publisher.
  - The spatial and temporal range of validity.
  - The intention of the publisher (why farmers should adhere).
  - The target audience (e.g. which products or farm types).
  - Procedures in the event of non-compliance.
  - A definition of terms used.
- A set of rules. Each rule may also have metadata attached defining in what cases it applies. Of these rules:
  - Some may be mandatory for compliance in all cases.
  - Some may be optional for compliance (a certain percentage must be met).
  - Some may be mandatory or optional depending on certain variables such as location, time, etc.

Each rule may be considered to be a logical predicate, i.e. a statement against which facts may be evaluated to produce a true/false answer. Usually each rule will either mandate an action ('the standard is complied with only if the farmer does *x*') or forbid an action ('the standard is not complied with if the farmer does *y*'). These may be defined as *obligations* and *prohibitions*. A further category of rule is the requirement to keep particular documentation. Such rules could be interpreted as an obligating rule where the action is the documenting, but since they should not affect the actual operations performed by the farmer, they are here treated separately. Rules may also not be explicitly defined as such – where only a text is available then individual rules must be inferred by interpreting this. This has been done for three example standards by FutureFarm WP2 to produce checklists of rules, which are presented in Vatsanidou et al (2009).

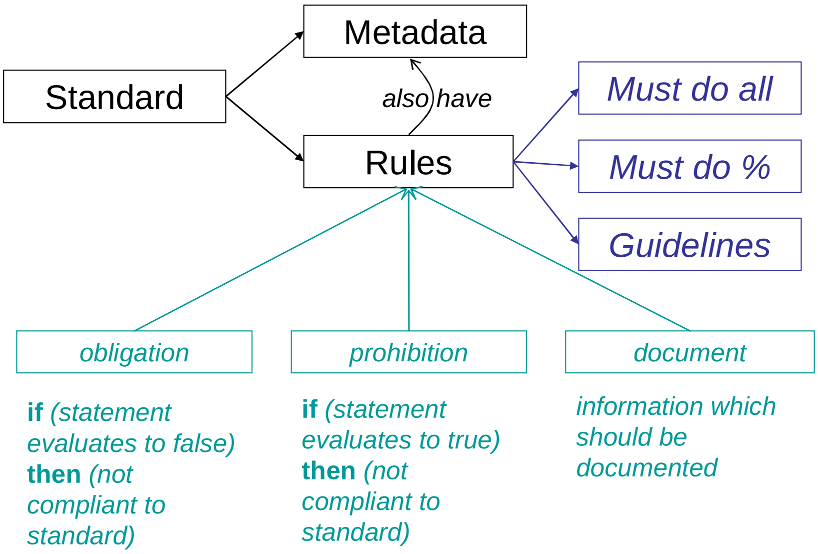


Figure 2. Structure of agricultural standards

In order for software to be able to automatically assess compliance to each rule, four fundamental prerequisites must be met:

1. *The rule must be encoded in a machine-readable form*. This may be hard-coded as algorithms in the software or be a transfer format (e.g. XML-based) which the software performing the assessment can read.
2. *The rule must be capable of being interpreted by the software*. This has two aspects. Firstly, all the concepts used within the definition of the rule (e.g. nouns such as *farm*, *production unit*, *crop*, *nitrogen* or *fertiliser*, verbs such as *weed*, *sow* or *spray* or adjectives such as *certified* or *organic*) can be correctly 'understood' by the software in the context in which they are used in the rule. Secondly, the rules must be computable, that is to say that it must be capable of being evaluated by an automaton (e.g. a Turing machine) according to computability theory.
3. *The individual rules must have discrete outcomes which can be determined by a computer*. That is to say that compliance to each rule must be assessed using computational models using digital data inputs and producing discrete outcomes (*true*/*false*) as opposed to value judgements.
4. *The required data inputs for assessment of the rule must be available in digital form*. This may be already existing data held in one or more databases which are accessible to the software or gathered on-demand by online sensors.

The first two of these prerequisites concern us here. The first is the background for this deliverable. The second defines the requirements of a transfer format, namely that we wish to be able to transfer computable rules and the definition of all concepts used within these rules. The former implies use of a transfer format for rules, and the latter a transfer format for ontologies (where an ontology is a "formal, explicit specification of a shared conceptualisation" (Gruber, 1993) or a definition of terms used and the relationships between them). The third of these prerequisites will be relevant in later stages of the FutureFarm project in which standards are expressed using the schema developed here and tested in prototype software. The final prerequisite may be expected to be one which will currently prove to be a barrier to use of automated assessment, but increasing update of informationintensive agriculture using e.g. precision farming and wireless sensors should in the long-term mean that the required data is increasingly available.

3. Use-Case Analysis

In this section we describe how the proposed encoding format is expected to be used. Based on this analysis, the design requirements for the format, presented in the next section, were determined. Note that this use-case analysis concerns the SOA-based system as a whole and not just the encoding format for agricultural standards. This analysis therefore also forms the basis of further work within WP4 of FutureFarm. The diagrams are presented using the UML notation.

3.1 Actors and Components in the system

Four actors are identified who are involved in the system. Their names, roles, and the organisations and people who are expected to perform this role are described in Table 1.

Actor	Role	Organisations
Catalogue Provider	The catalogue acts as a clearinghouse/search-engine for standards and repositories. Using the system of interlinked catalogues the FMIS may locate appropriate servers from which standards will be accessed. Each catalogue should contain metadata for a number of other catalogues and repositories. The catalogue provider will probably therefore be a semi-independent or umbrella organisation, not an organisation which directly publishes standards.	Government agencies, advisors, unions, collectives, ...
Standards Publisher	The organisation that defines the content of the standard. Note that they may not actually provide the standard themselves via a web-service/repository: this task may be contracted to a specialist organisation (which may provide this service for more than one standards publisher)	EU, national and regional governments, industry groups (label organisations)
FMIS	Tool used by the farmer to manage the farm, and in this case to find and use information related to standards	May be integrated in current FMIS (desktop or web-based) or a separate web-application for simple search-and-display functionality.
Farmer	Needs information about standards in order to correctly manage the farm. Could also be a farm advisor and not a farmer directly	Farmer, farm advisor
Autonomous farm machines	(Semi-)Autonomous farm machinery (e.g. robots) may require information about standards in order to implement an operational plan correctly and according to the standards. In terms of the use cases identified and presented here, autonomous farm machines may be considered as having the same role as the FMIS (client software) and/or the Farmer (actual decision-making)	Farm machinery

Table 1. Actors in the proposed system

The software components which have been identified and which are shown in the use-case analysis are described in Table 2.

Component	Role	Notes
Catalogue	Delivers metadata covering standards and repositories (servers) serving them.	It is not expected that a single catalogue holds all information about everything - a cascading system is envisaged whereby a catalogue may also be used to find further catalogues which hold information not held by the initial catalogue. There are therefore multiple catalogues in the complete system.
Server (repository)	Delivers information concerning standards. This information is the metadata describing the standard (also available via the catalogue) and the definition of the actual rules which must be followed to be compliant to that standard.	Each repository may hold information on one or more standard, and multiple repositories may be available.
Rules Application	Takes a rule definition and assesses compliance based on data from the FMIS and/or external sources (local filesystem, web services, wireless sensors, etc.)	The rules application may be a implemented as part of the FMIS (either monolithic or as a software module) or as an external service (e.g. a web service).
FMIS	Client software which can query catalogues and repositories (based on certain parameters) to find further catalogues and repositories as well as to retrieve the definition of the individual standards, or even of individual rules which form part of those standards. The part of the FMIS responsible for managing interactions with Rules is referred to here as a Rules Manager.	More than one FMIS should be implemented within WP4 - a basic, web-based one which provides basic search-and-display functionality and one embedded within an existing FMIS which will be able to evaluate an operational plan against the standards.

Table 2. Software components in the proposed system

3.2 Overview of Use-Cases

Figure 3 shows an overview of all use-cases considered for the system. In principle, the full use of the system can be considered as a series of steps:

1. Using catalogues to locate further catalogues holding metadata about available servers.
2. Using servers to retrieve metadata about individual standards.
3. Viewing information about standards and configuring the FMIS to respect chosen standards.
4. Using servers to retrieve the rules which are relevant for a particular context (e.g. planning or evaluating a particular operation such as fertilisation).
5. Testing a plan against rules retrieved from the servers using a rules application.

The first 3 should only be necessary on an occasional basis. Steps 4 and 5 will however be regularly repeated. The encoding to be developed here will be used for transfer mainly in steps 3, 4 and 5. Note that in this diagram the separation between the FMIS and the Rules Application component is not considered – i.e. a monolithic system is assumed in which the FMIS incorporates the Rules Application.

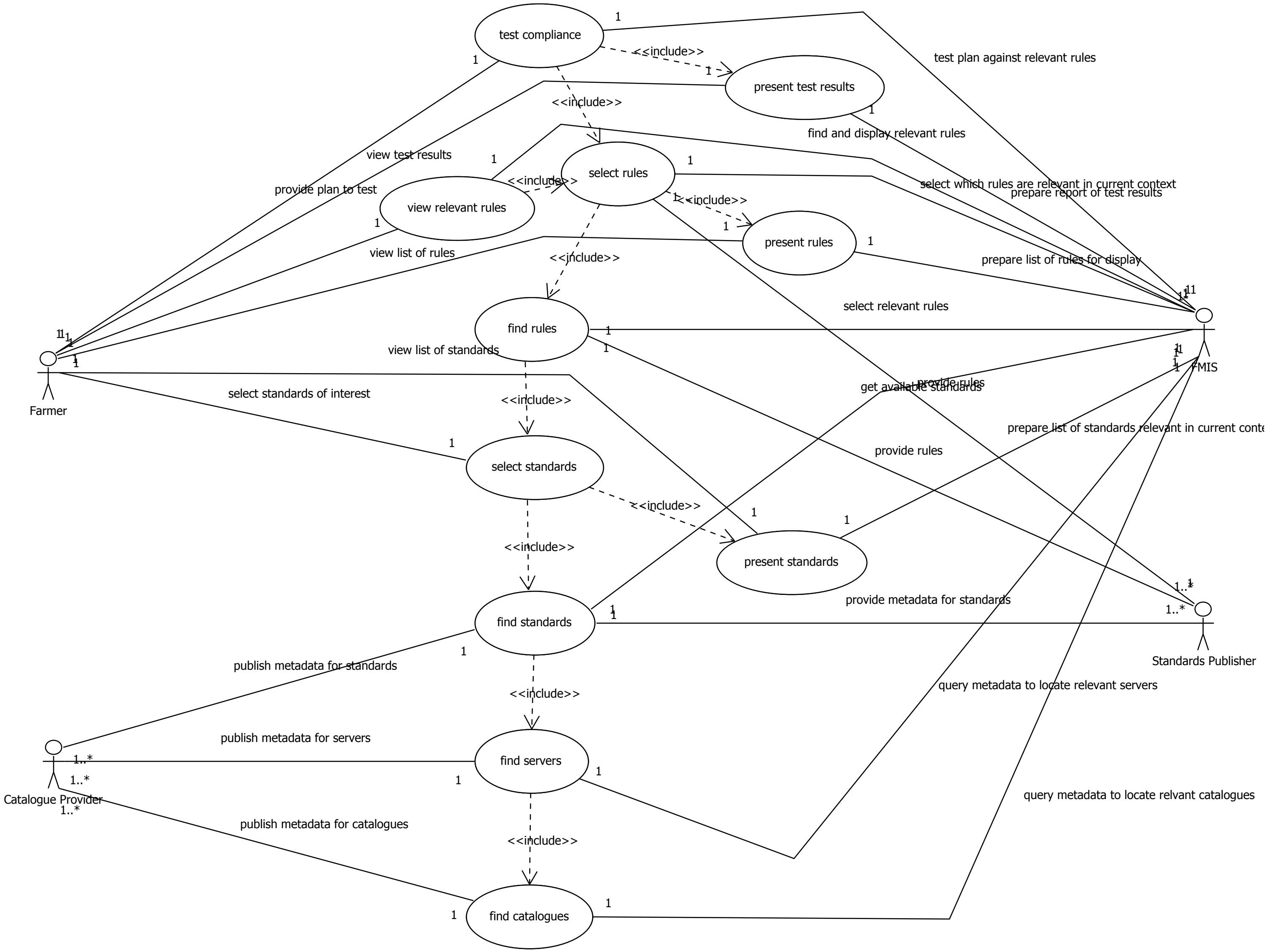


Figure 3. Overview of all use-cases identified for the proposed system

3.3 Software components

Figure 4 indicates the expected structure of the FMIS software. Two models of interaction between the FMIS/Rules Manager and the Rules Application are shown; 'push' and 'pull'. In the first case, the FMIS identifies required data from the database and pushes this to the Rules Application. In the latter case, the database provides a standard interface which the Rules Application may use for querying to retrieve the required data.

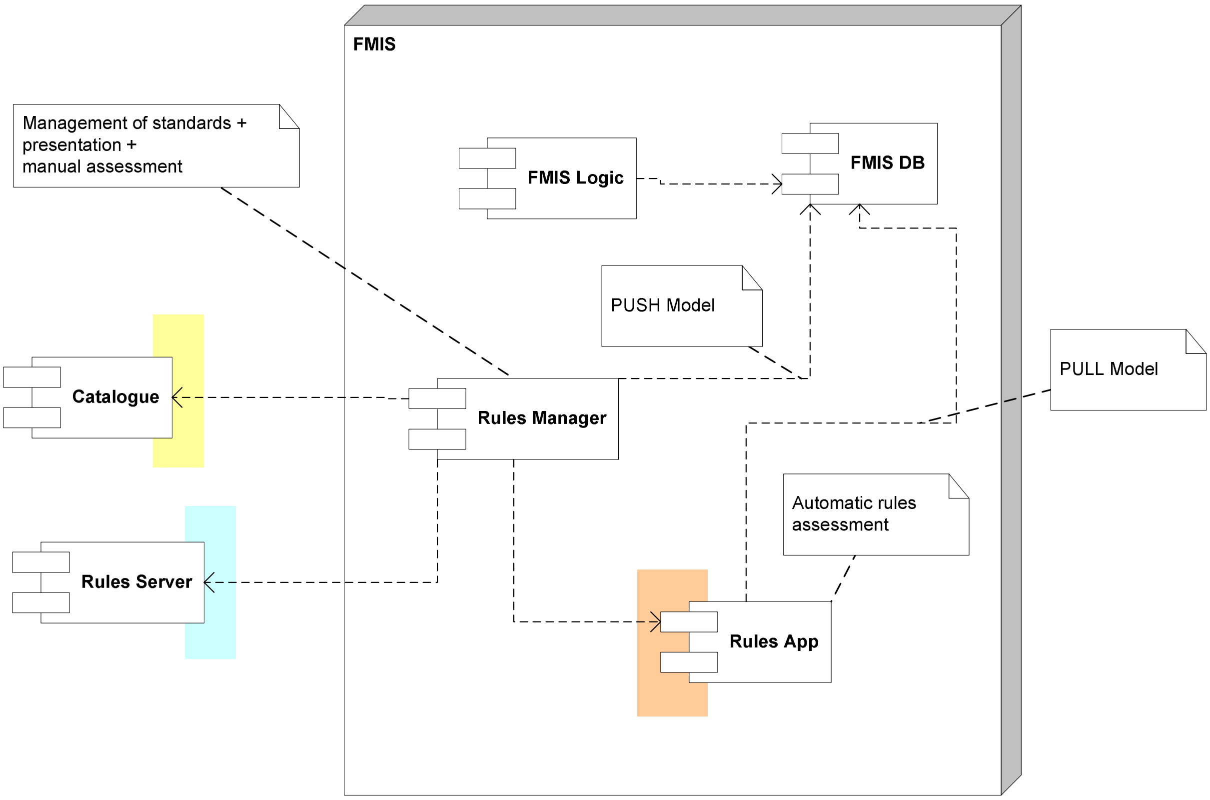


Figure 4. Interactions with the Rules Subsystem in the FMIS

4. Design Requirements

It is assumed that the encoding schema to be developed will be generated based on a data model which represents the required information. As such, the design criteria are split into three categories; for the data model, for the vocabulary and for the encoding schema itself. The criteria were identified based on the concrete case of fertilisation according to organic farming based on Finnish regulations, but will also be transferable to the general case.

The identified criteria are based on the information required for an automated validation of a plan against a set of rules. The amount of information is far from trivial and contains several layers of abstraction within it. As an educated guess, it can be assumed that this complexity does not form a problem for the encoding schema or the vocabulary apart from the apparent increase in the size of both. The data model, however, becomes the immediate problem. The encoding of the data model or the vocabulary used in the encoding are likely to remain simple, but increase in size as the amount of the required "simple" information increases. As a major increase in complexity compared to the one presented in a use-case, the actual validation process requires additional information besides the transient and in-season plans and the rules themselves. This additional information composes of general farm information and the historical information of the farm.

4.1 Criteria for the Data Model

The data model is the most complicated part as it has the implicit criterion of being able to represent all the information required in the validation process. The encoding schema should initially apply only to the rules used in the validation. The data model contains elements common in agriculture and likely to be found in existing standards or emerging standards. Especially for these common elements, something like AgroXML should be considered. For all GIS data, an existing GIS data model should be used.

The data model should be able to present the following rules

- Existential
  - Positive (something must exist)
  - Negative (something must not exist)
- Amount
  - Over space
  - Over time
- Origin (for manure)
  - Farm and the farming method if several are used by the farm
  - Animal
- Optionality
  - Complete optionality (rules must not be adhered to)
  - Limited optionality (some number of rules must be true)
- Method of application
- Machinery (can be abstracted to a list of allowed machinery and specifications)
- Crop cycles
  - Field, crop, timeframe (these should not be a problem)
- Permissions (make something illegal legal)
- Agreements (additional rules to adhere to (additional rules))

4.2 Criteria for the Vocabulary

The criteria for the vocabulary are straight-forward as the vocabulary needs to cover all the information and the rules required for the validations process. Use of an existing basic agricultural vocabulary, such as AGROVOC (FAO, 2009), should be considered. The criteria for the content of the vocabulary are the same as the information required as well as the types of the required rules. Some considerations for the vocabulary:

- Interoperability between the rules of different organisations. This is problematic unless all organisations agree to represent their rules using a common vocabulary, but since no formal encoding of rules for agricultural standards exists, this is theoretically possible and is a possible assumption to reduce complexity, although probably unrealistic in the longer term.
- Use of an existing vocabulary either as a base or together with an additional vocabulary.

4.3 Criteria for the Encoding Schema

The encoding schema should be powerful enough to represent at least Horn clauses[2]. Alternatively the schema can support entire first-order logic or higher order logic. However, using a level of logic more expressive than Horn clauses can have negative computational

[2] "A clause is called a Horn clause if it contains at most one positive literal… Horn clauses express a subset of statements of first-order logic. Programming language Prolog is built on top of Horn clauses." (Sakharov, 2003)

effects or result in a non-computability. For actual inference, a decision between a closed-world and an open-world has to be made.

The encoding schema should be human readable for the sake of development and debugging as encoding efficiency should not be an issue – some form of XML is preferable. There are some existing encoding schemas for logical rules such as RuleML which is powerful enough to express the Prolog programming language and hence Horn clauses. The encoding schema should be kept to the bare minimum of what is required as additional functionality should be an addition to the schema instead of a change.

5. Existing Standards for Machine-Readable Encodings

In this section we will briefly present the results of a literature review of machine-readable encodings. These are broken-down into four areas, namely for agricultural data, metadata, rules and ontologies. Although standards for agricultural data do not necessarily play a direct role in an encoding of agricultural standards, they base implicitly on a particular data model, or ontology, of agricultural data and contain mechanisms for referencing products used or produced in agriculture (fertilisers, plant protection, crop types) which may be re-used.

5.1 Agricultural Data

Five standards for transfer of agricultural data have been identified. With the exception of ISO17138, all of these standards are still under development. The extent of their use in practical applications is largely unclear.

5.1.1. ISO11783 (ISOBUS)

ISO11783-10 defines a communication format between farm management software and an on-machinery controller. Two elements of this are potentially of interest here – a model of geometry for on-board devices and an XML-based transfer format. The latter presents a model for operational data based on a strongly entity-relationship model and uses English-language XML element names abbreviated for compactness. ISO11783-11:2006 specifies a data dictionary of identifiers for process data variables or data elements used in ISO11783-10.

5.1.2. AgroXML

AgroXML is developed by KTBL (Association for Technology and Structures in Agriculture) in Germany together with a consortium mainly composed of agricultural software suppliers and is currently in version 1.4. AgroXML itself is English-language and consists of a modular XML schema for representing data and a number of content lists to provide constrained content for particular elements (e.g. crop growth stage, pesticide active ingredients, seeding method, etc.). AgroXML is largely object-oriented in its modelling approach, and is becoming increasingly so. The scope of AgroXML currently covers all arable farming, with extension to cover animal production currently in development.

5.1.3. AgXML

AgXML is an effort to develop standards for e-commerce in the American grain and oilseed industry and is being developed by a consortium of grain industry companies. AgXML uses a document-based modelling approach, whereby some elements are shared between document types.

5.1.4. eDAPLOS

eDaplos is currently a largely French-led project, although being developed in English-language. eDaplos has specified an XML-based representation of a crop data sheet within the framework of the UN/CEFACT ebXML. The modelling methodology is message-based, with the data modelling apparently based on legacy formats.

5.1.5. EDI-Teelt

EDI-Teelt is a Dutch-language XML schema. Similar to eDAPLOS, the modelling methodology is message-based, and apparently based on legacy formats.

5.1.6. Summary

Of existing and emerging standards for agricultural data transfer, none has yet gained a large market share, and most are largely national efforts. The exception to this is ISO11783 (ISOBUS), although this is targeted at the transfer of data from the (farm management) software to on-board devices and not transfer of data between software systems and organisations. In terms of how generally applicable the model is and the reusability in the current context, the ISO11783 data dictionary and the AgroXML content lists may both provide useful resources for specifying particular items. The object-oriented modelling approach of AgroXML may also provide a basis for specifying ontology.

5.2 Metadata

Four standards for metadata have been identified. It should be noted that Dublin Core defines both the metadata to be recorded and the format, whereas ISO19115 and ISO19119 define a metadata model whose encoding schema is defined in ISO19139.

5.2.1. Dublin Core

"The Dublin Core Metadata Element Set is a vocabulary of fifteen properties for use in resource description… The fifteen element 'Dublin Core' described in this standard is part of a larger set of metadata vocabularies and technical specifications maintained by the Dublin Core Metadata Initiative (DCMI)" (DCMI, 2004). The Dublin Core metadata elements were accepted as ISO15836:2003. They allow a very generic description of any resource (typically used for webpages), including title, publisher, date, subject, language, contributors, resource type, spatio-temporal coverage, etc. The actual value of each element is generally unconstrained, such that interpreting the content automatically may be problematic.

5.2.2. ISO19115

ISO19115:2003 aims "to provide a structure for describing digital geographic data,… defines metadata elements, provides a schema and establishes a common set of metadata terminology, definitions, and extension procedures." (ISO, 2003). ISO19115 applies to the full description of datasets, dataset series, and individual geographic features and feature properties. A minimum set of metadata is defined, together with optional elements and extension points. The metadata model bases on the model of geographic information developed in the ISO19100-series standards. As well as basic metadata (identification, description, origin, actuality, usage, extent) as is included in Dublin Core, additional information regarding data quality, spatial representation and reference system, portrayal and application schema may be recorded.

5.2.3. ISO19119

ISO19119:2005 describes a taxonomy of geographic web services and both platform-neutral and platform-specific service specifications. In particular, the metadata relevant needed for such services is specified.

5.2.4. ISO19139

ISO19139:2007 provides an XML schema for encoding of the ISO19100-series metadata standards ISO19115 and ISO19119 allowing metadata for geographic information and geographic web services to be represented in machine-readable form for transfer between systems.

5.2.5. Summary

ISO19115 (encoded using ISO19139) allows a much more in-depth and formalised description of datasets than Dublin Core. However, due to its greater simplicity, Dublin Core is more widely used in practice, and the ISO 19100-series standards have until now only been used within the geoinformation community. For general metadata, the Dublin Core elements are probably therefore preferable, but where more precision is required, the ISO19100-series may be used.

5.3 Rules

The field of rule markup is still a developing research area and there are no well-established standards. Five current or recent initiatives have been identified and are described below – it is worth noting that many individuals are or have been involved in more than one of these initiatives.

5.3.1. RuleML

The Rule Markup Language (RuleML) is an XML based markup language for the representation of rules. RuleML itself is only for the representation and exchange of rules without considerable consideration to features beyond those. Several markup standards are currently based on RuleML.

5.3.2. R2ML

The REWERSE Rule Markup Language (R2ML) is a language intended for rule interchange and reasoning. Like most other languages, R2ML is XML based and implements a relatively standard set of features for a rule language. Features of R2ML include a formal XML schema for validation, an abstract syntax based on EBNF, differing semantics for rules and support for rules of integrity, derivation, production and reaction.

5.3.3. SWRL

Semantic Web Rule Language (SWRL) is a combination of the languages OWL and RuleML. SWRL retains the full power of OWL DL but at small practical costs including decidability. SWRL bears resemblance to logic programming with Horn clauses and has a relatively human readable syntax in addition to the concrete XML syntax. Unfortunately, since SWRL is not decidable, no implementation supports the full SWRL specification.

5.3.4. RIF

Rule Interchange Format (RIF) is an emerging W3C recommendation for the encoding and interchange of rules. As an emerging recommendation, RIF is still work in progress with few if any tools existing for it. RIF has the expressive power of Horn clauses without function symbols.

5.3.5. LKIF

Legal Knowledge Interchange Format (LKIF) is a language composed of several sublanguages such as RDF and OWL. LKIF is intended for the encoding of legal knowledge with an emphasis on portable exchange of legal knowledge. While interesting in itself and as a composition of existing languages, LKIF is domain specific to legal knowledge.

5.3.6. Summary

Decidability and domain-independence are key requirements in this case, and so SWRL and LKIF are not suitable. RuleML and R2ML appear to no longer be being actively developed, whereas RIF is being actively developed within the W3C. Although tools for handling RIF are not widely available, this may be the best long-term option for rule markup.

5.4 Ontologies

Two strongly-linked languages for representing ontologies have been identified, RDF and OWL. They are both recommendations from the W3C and are widely used in research, and to a lesser extent in practice.

5.4.1. RDF

Resource Description Framework (RDF) is a family of specifications intended for the modelling of metadata. RDF is an abstract model based on triplets of subject, predicate and object, therefore RDF has several serialisation formats of which two are currently in common use. RDF is intended for automated processing and there exists query and inference languages for RDF of which SPARQL is currently dominant. RDF schemata are specified by RDFS, which is a knowledge representation language for specifying RDF vocabularies (ontologies).

5.4.2. OWL

Web Ontology Language (OWL) is a family of extension to RDFS intended for the definition of ontologies. OWL retains the functionality of RDFS and three levels of OWL exist. OWL Lite is the most simple implementation of OWL and is restricted to simple classification tasks, OWL DL is the compromise between expressiveness and computability, and retains the computation power of description logics. OWL Full is the most expressive version of OWL but without any guarantees on computational efficiency and it is unlikely that any reasoning engine will ever fully support reasoning for OWL Full. OWL is commonly serialised to RDF/XML syntax which makes automated handling of OWL ontologies feasible with existing tools. A plain XML schema (without RDF) is also available for OWL, which is also compatible with many tools.

5.4.3. Summary

Due to the greater expressiveness and availability of an XML schema, which allows instance documents to be validated, OWL is probably the preferable language. To retain computability, only the OWL DL subset should be used.

6. Data Model

In order to provide a clear documentation of the transfer format, the data to be encoded has been modelled using a UML static structure model. This is shown in Figure 5.

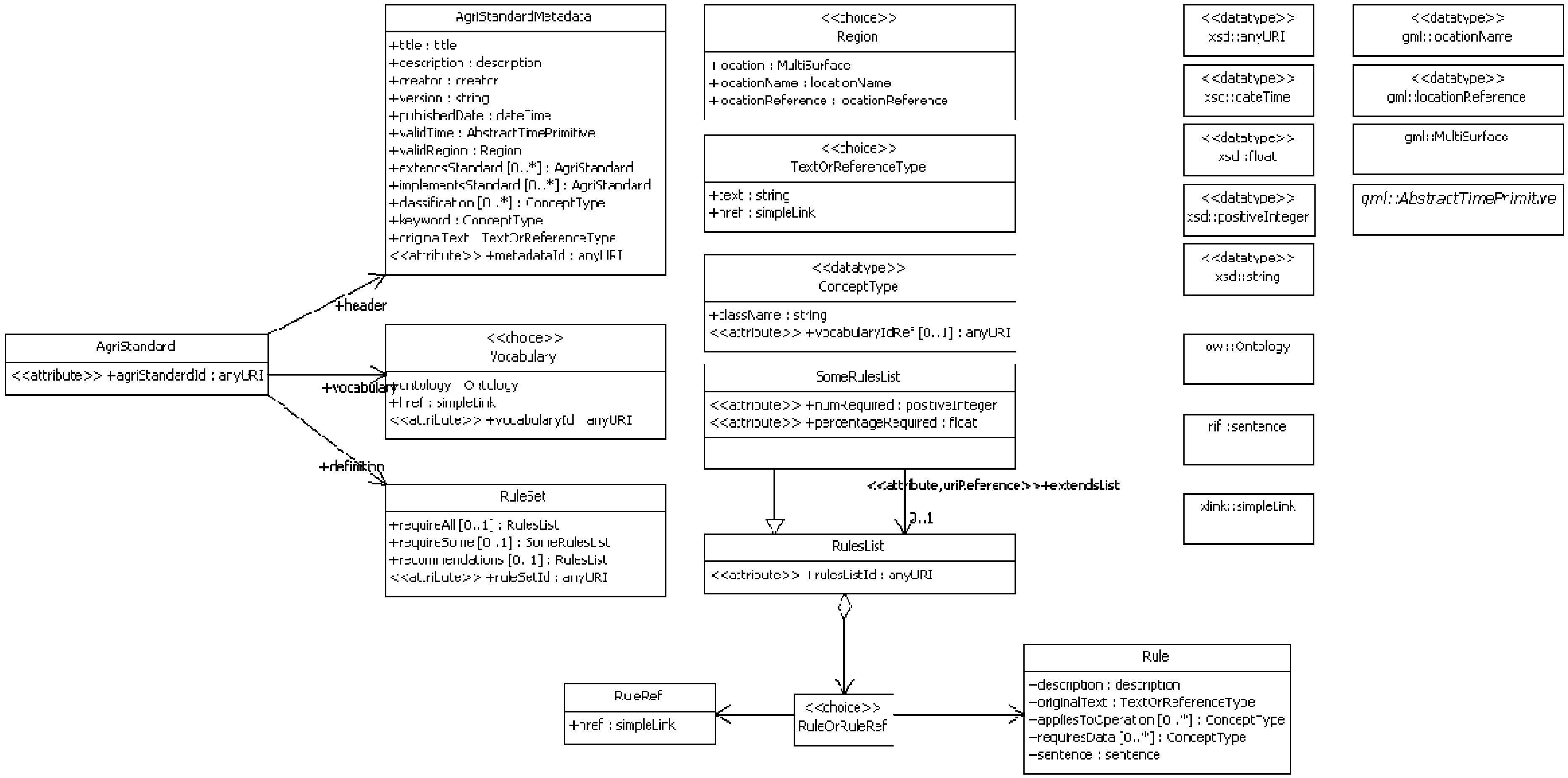


Figure 5. UML static structure model for the encoding schema

7. XML Schema

The data model shown in [Figure] has been converted to an XML Schema, which is illustrated in Figure 6. The prefixes to the element names indicate the schema in which the elements are defined; dc for Dublin Core, gml for Geography Markup Language, owlx for Web Ontology Language, rif for the Rules Interchange Format and ff for locally-defined elements. The full schema is stored in the project version management system[3] and will be modified as necessary as experience is gained while implementing the required server and client software. The final schema will be made publically available once it has reached a stable state.

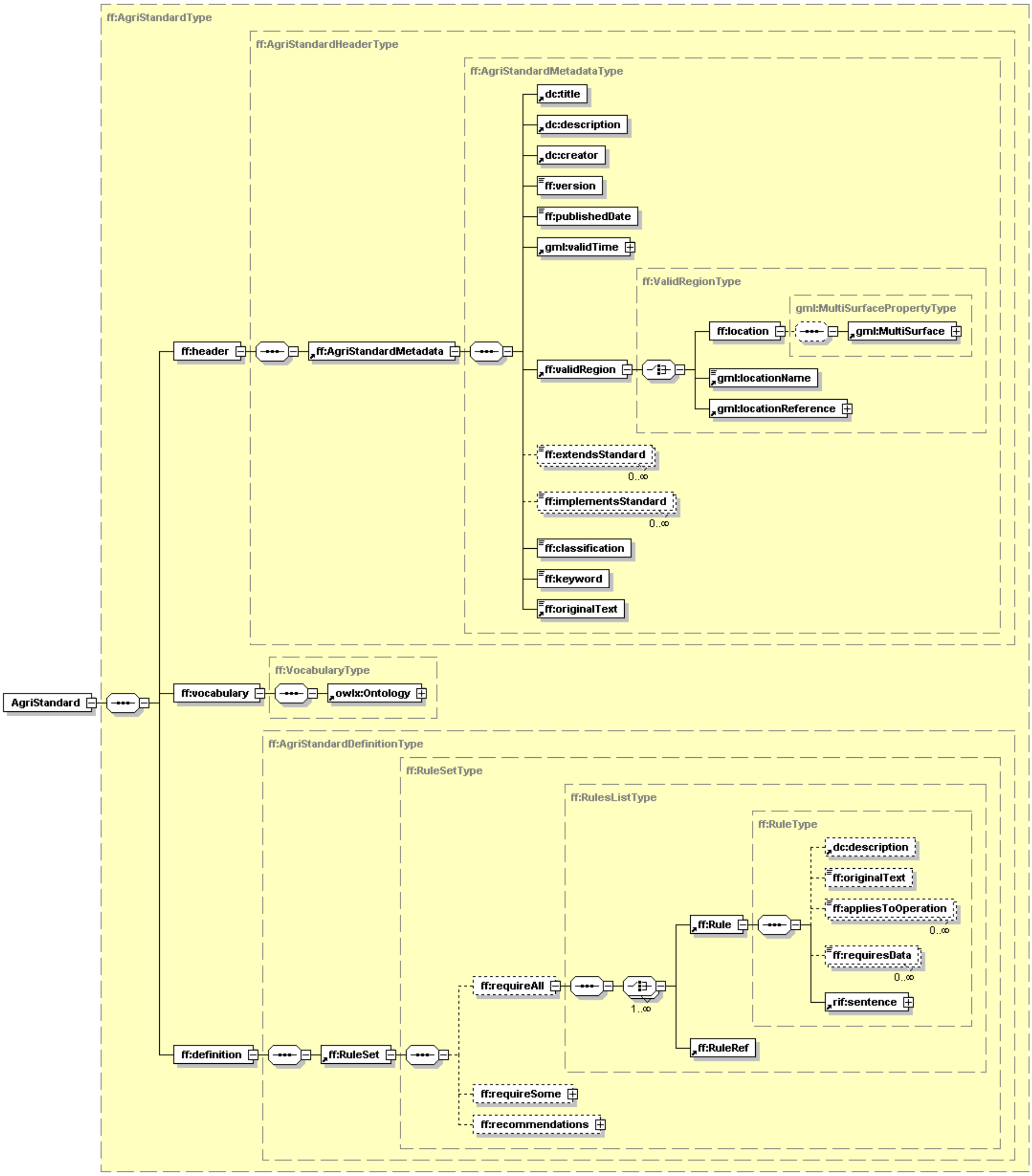


Figure 6. The structure of the XML Schema developed illustrated using the software 'XMLSpy'

[3] https://svn.futurefarm.eu/schema (username and password required)

References

Boer, A., Di Bello, M., van den Berg, K., Gordon, T., Förhécz, A., Vas, R., 2007. *Specification of the Legal Knowledge Interchange Format*. ESTRELLA Deliverable 1.1. Available at: http://www.estrellaproject.org/doc/D1.1-LKIF-Specification.pdf (accessed 2009-02-26).

DCMI (Dublin Core Metadata Initiative), 2004. Dublin Core Metadata Element Set, Version 1.1: Reference Description. Available at: http://www.dublincore.org/documents/dces (accessed 2009-02-27).

FAO (Food and Agriculture Organisation of the United Nations), 2009. AGROVOC Concept Server. Available at: http://www.fao.org/aims/agrovoccs.jsp (accessed 2009-02-26).

Gardner, A., 1987. *An artificial intelligence approach to legal reasoning*. MIT Press.

Gruber, T., 1993. A translation approach to portable ontology specifications. Knowledge Acquisition 5 (2) 199-220.

ISO (International Organisation for Standardisation), 2003. Geographic Information – Metadata. ISO19115:2003.

Sakharov, A. Horn Clause. From *MathWorld*--A Wolfram Web Resource, created by Eric W. Weisstein. Available at: http://mathworld.wolfram.com/HornClause.html (accessed 2009-03-17)

Vatsanidou, A., Fountas, S., Aggelopoulou, A., Gemtos, F., Strobl, K., Charvat, K., Pedersen, S.M., Sørensen, C., Nash, E., Blackmore, S., 2009. *Compliance to standards specifications*. FutureFarm Deliverable 2.1.1.