TNO
Capitool 10
7521 PL Enschede

www.tno.nl

T +31 88 8662450
F +31 88 8662147
info-ict@tno.nl

TNO-rapport

Rapport nr TNO 2013 R11088

Semantics for Smart Dairy Farming: a milk production registration standard v1.0

Date	17 july 2013
Authors	Silja Eckartz, Jack Verhoosel, Erwin Folmer, Erwin Somers
Documentnumber	
Nr of copies	
Nr of pages	44
Nr of appendices	0
Client	SmartDairyFarming consortium
Projectname	SDF
Projectnumber	052.03031

Table of contents

1 Introduction 3

2 Scope of the standard 4
2.1 The insemination monitor as starting point 4
2.2 Focus of the standard on milk production data 7

3 Model-based interoperability approach 8
3.1 MOSES methodology and interoperability 8
3.2 The model-based approach 8

4 Common Domain Model for milk production data exchange 11
4.1 Actors and their roles in the domain 11
4.2 Objects in the domain 12
4.3 Events in the interaction between actors in the domain 13
4.3.1 Event: animalDataList 14
4.3.2 Event: registerMilkingResults 15
4.3.3 Event: exchangeSampleInformation 15

5 Common Information Model for milk production data exchange 16
5.1 Information objects and their relation 16
5.2 Information elements 17
5.2.1 Animal identification 17
5.2.2 Milking result 19

6 Common Message Model for milk production data exchange 24
6.1 Actual Initial Herd list message 24
6.2 Transfer In message 24
6.3 Transfer Out message 25
6.4 Milking results recording message 25
6.5 Sample message 27

7 Technical Model 28
7.1 Use Cases 28
7.1.1 Event: animalDataList 28
7.1.2 Event: registerMilkingResults 29
7.1.3 Event: exchangeSampleInformation 30
7.2 Messages 32
7.2.1 ComplexTypes 32
7.2.2 Standard Response messages 37
7.2.3 initialHerdListRequest and initialHerdListResponse 37
7.2.4 transferAnimalRequest and transferAnimalResponse 39
7.2.5 milkingResultRequest and milkingResultResponse 41
7.2.6 provideSampleRequest and provideSampleResponse 42
7.2.7 sendSampleDataRequest and sendSampleDataResponse 42

1 Introduction

This document describes a standard for the information exchange between different actors in the milk chain. The scope of this standard is the exchange of milk production data between milking equipment on the farm and milking recording organization that can be used to determine the best insemination moment of an individual cow.

We used the MOSES method to develop the standard [TNO 2012 R10544]. This is a method to develop semantic standards using a model-driven approach. The interface to cooperate is hereby specified using models that define how information is exchanged between different actors. MOSES is a good way to specify the semantics of both the information as well as the processes. Thereby, an explicit distinction is made between the semantics of the information and processes in the business domain on the one hand and the actual technical way of exchanging information on the other hand. This separation gives the developer of the standard the freedom to not specify the way the information is exchanged within the standard and thus it is easier to change between different ways of exchanging information, such as messages, web services, e-mail, linked open data. This document follows the steps described in the MOSES methodology.

In chapter 2, the scope of the standard is defined in more detail. Next, in chapter 3, the MOSES methodology is described in an overview, in which a common domain model, a common information model, a common message model and a translation to a specific solution are defined. In chapter 4, the common domain model is described that defines the actors, roles and systems in the domain as well as the most important objects and events in the context of the defined scope. In chapter 5, the common information model is derived in which an overview diagram is presented with the main information objects and a detailed description of the information elements is given. In chapter 6, the common message model is defined that describes the messages that need to be exchanged between the actors in scope about milk production data as part of milking results. In chapter 7, a technical mapping of the common information model and the messages towards a technical implementation based on XML is given.

2 Scope of the standard

The standard described in this document has been prepared under the Smart Dairy Farming project. Within this pilot project, companies, research institutes and farmers work together to develop innovative means, which farmers can use to prolong the life of cows. The result consists of sensors, indicators, decision models and advice that help the farmers in the care of their cows. Making the right choices in daily operations contributes to better health and longevity of the animals and thus to a more sustainable supply chain.

Dairy farmers make choices based on their experience and technical business information. Partly due to the growth of farms, steering is generally applied at torque level. However, this is not always accurate enough to care for individual cows optimally. Moreover, the needs of individual animals are not always visible. Control at the level of the individual animal is necessary to lengthen the productive life of cows for the dairy farmer. Sensors, indicators and decision models can help dairy farmers in identifying needs and making the right choices.

Smart Dairy Farming focuses on the critical factors animal health, fertility and feeding. Using among others sensors, data on individual animals is collected. This data forms the input for decision models, indicators and advisory products that can be applied in daily practice. With these tools, dairy farmers can make choices in the care of individual animals. The three areas each have their own objectives and require different types of data.

- Animal health: (1) optimization cattle-rearing, to lay the basis for long-lived cows: data on the development of body weight and milk intake; (2) optimization of the dry period to get animals healthy into production; data on rumination, feed intake, body condition score and behavior.
- Fertility: optimization calf pregnancy results, to prevent unnecessary downtime data on heat detection and the optimal insemination moment.
- Feeding: optimization rumen condition and metabolism by individual balanced nutrition and care in order to counteract acidosis: rumen acidity and milk urea content.

2.1 The insemination monitor as starting point

Within the Smart Dairy Farming project as part of the objective fertility one of the use cases focuses on the design of an insemination monitor. With this monitor, it is possible to get a better understanding when it is the best time to inseminate a specific cow. The purpose of the insemination monitor is to make better decisions regarding the insemination moment. Intervention gives among others better, combined data to support the decision. The input to the monitor consists of animal data (performance, milk production, sensor data, cow activity), business information and economic parameters. It is expected that the insemination monitor can make use of existing information flows, but that they will become more intensive and contain more data.

The background of the insemination monitor is that the cow usually has a one-year cycle of calving and most of the farmers try to inseminate each cow once per year. A calf is carried for about 9 months and 2 months after the calf is born the cow can be inseminated again. From that moment on the cow has a 3-weekly fertility period in which she can be inseminated. After the first calving, a cow starts to produce milk. In principle a cow can be milked twice per day in the 10 months after a calf is born. Therefore, from a financial point of view, a farmer wants a cow to calf once per year.

However, it can be more efficient to use much more available information for determining the exact moment of insemination in order to maximize the birth of healthy and strong calves. Therefore, there is a tendency towards more calfs per cow then the once-per-year cycle. In addition, more information on the specific cow becomes available that can be used to better judge whether insemination at a certain moment is best from various combined perspectives.

There are various stakeholders that have an interest or a specific role in the insemination monitor (see Figure 1).

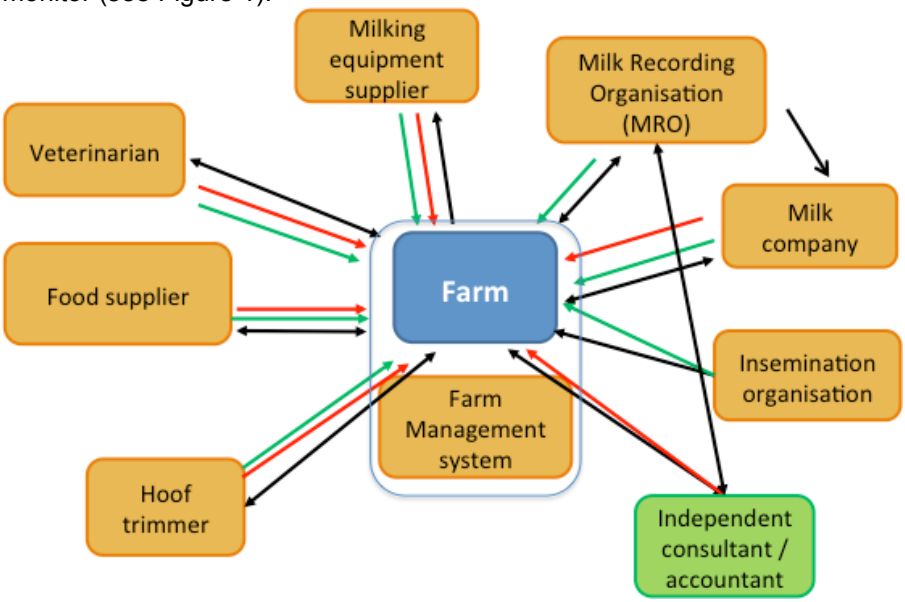


Figure 1: Stakeholders for the insemination monitor.

In another working group of the SmartDairyFarming project, the various stakeholders have been inventoried and studied. The figure above presents these stakeholders and their relations with respect to the insemination monitor. The black arrows represent information, the green arrows represent services/products and the red arrows represent consultancy. For each of the stakeholders an indication will be given below on the advantages an insemination monitor can have.

In general, the impact of the insemination monitor is that it adds to a faster genetic progress in livestock and longer life of livestock. In addition, the monitor can also have impact on the feed balance per cow and on the execution flow of cows.

With the insemination monitor, the farmer can make a choice whether or not to inseminate in a different and more decision supportive way. In that sense, the tool gives and advice and the farmer decides. The farmer wants support for the yes/no question, for example:

- No, because chances of pregnancy is 60% (the farmer can still take his chances with a straw of 5 euros of course)
- No, because this cow was sick a lot lately
- No, because this cow has often given no/bad calf (maybe he also wants to know if he should not inseminate anymore)
- No, not now, but maybe next trip

If the farmer inseminates at a "prior probability" of e.g. 60% (the first bullet), the result (healthy calf or not) can also be returned to the farmer and the insemination organisation in order to learn. All in all, this leads to a faster genetic progress of the livestock and a longer livestock life.

By using the insemination monitor, the insemination organisation receives new data on existing arrow, namely the advice and explicit choice of the farmer that the cow will be inseminated. Currently, the organisation only hears if a straw is ordered. In addition, another way of bull advice is incorporated when it turns out that as some cows give no or poor calves.

The milking equipment supplier can extent the milking robot functionality with additional sensors. The farmer may choose to earlier replace the robot with additional functionality. The supplier may also provide separate units to another brand. Last but not least, the supplier can provide real-time milk production data to determine the right insemination moment and in return can receive aggregated data on milk production and inseminations.

The impact on the milk recording organisation is that it can receive and provide real-time milk production data and can better develop inline content measurements. The organisation can develop and provide their own insemination monitor based on the entire set of information that is maintained.

The independent consultant/accountant, can give the farmer insight into whether it is wise to expand (or shrink) the herd. This can be done by calculating company-specific standard amounts on rearing, expected feed balance and expected production-life balance.

The impact for the food supplier is that the monitor will contribute to a better determination of the feed balance and feed efficiency for the cow. Thereby, the supplier can provide a better amount and mix of food.

For the milk company, the insemination monitor is not relevant for the short term. However, for the long-term planning it can become important in order to estimate how much milk will be delivered. Especially in completely quota-free era the milk company can benefit from better planning of the milk production.

The impact for the hoof trimmer is that the insemination monitor can provide more extensive diagnostic and measurements. In return hoof trimming data regarding illness of the cow can be used by the monitor.

In summary, the main advantages of the insemination monitor are:
- Increased longevity and production of lifestock by better decision making on insemination.
- Farmer knows better and more precisely when he should inseminate a specific cow in the 3-weeks fertility period.
- Milk recording organisation and food supplier get real-time data from a specific cow can based on that give a better advice for insemination.
- Milking equipment supplier has to provide real-time milk production data and partly also food intake per cow and receives in return aggregated milk production data per cow and farm.

We assume that the insemination monitor can in principle be organised and operated by any stakeholder in the milking supply chain. Nevertheless, it needs information coming from various sources: the milking equipment, food intake equipment, pedometers and other sensors at the farm that keep track of the behavior of the cow.

In general, the following data is needed by the insemination monitor to make a good estimation of the right insemination moment:
- Daily milk production
- Real-time content in milk including SCC

- Current market prices (milk, meat, food), company-specific standard amounts for rearing costs
- History of the animal: disease, fertility, production, claw (= illness)
- Food balance and feed efficiency
- Condition score.
- Degree of genetic progress (= genetic potential of animals in breeding)
- Animal disease (paraTBC, Salmonella)

2.2 Focus of the standard on milk production data

After a discussion about the various types of information to be used for the insemination monitor, it was decided to first focus on the exchange of real-time milk production data and milk content. The reasons for this focus are:

- There are internationally multiple suppliers of milking equipment that produce milking production data and also multiple milk recording organisations that collect and maintain this data. Thus, as standard for this m-to-n information exchange is useful.
- There is already information exchange between the farm and the milk recording organisation, but not on a daily, real-time basis. Thus, the basics for a standard in this area are already measured and available and the implementation should not be to difficult.
- The use of milk production data seems to be one of the most important factors for the insemination monitor. Thus, standardisation of this information will enable better insemination monitors to be developed by multiple stakeholders along the chain.

Therefore, in this document we concentrate on the standard for the exchange of information between the milk equipment on the farm and the milk recording organisation that keeps track of herds and production data. For now all other information (e.g. on food intake, economics etc.) is out of scope. At a later stage, the standard can be extended with other types of information, but the focus is to start with a small, core standard.

Nevertheless, the common information model that is described in this document also presents an overall view on all the information components that are needed for the insemination monitor. Subsequently, we will then focus on the milk production information elements and the messages needed to exchange this information. Before describing all this, we will first shortly present our model-based approach to define the domain, the information model and the standard messages.

3 Model-based interoperability approach

This chapter introduces our approach for the development of interoperability solutions on the basis of a good model of the business domain. Our methodology is called MOSES, Model-based development of SEmantic Standard. We briefly describe the perspective of our methodology and the relation with interoperability. In addition, we define our business domain approach and the type of models we develop in order to get towards a set of standard messages to be used as interoperability solution.

3.1 MOSES methodology and interoperability

To achieve interoperability between organisations cooperation has to be established at different levels. We use a standard 4-layer architecture that is common in the area of business IT system (Figure 2).

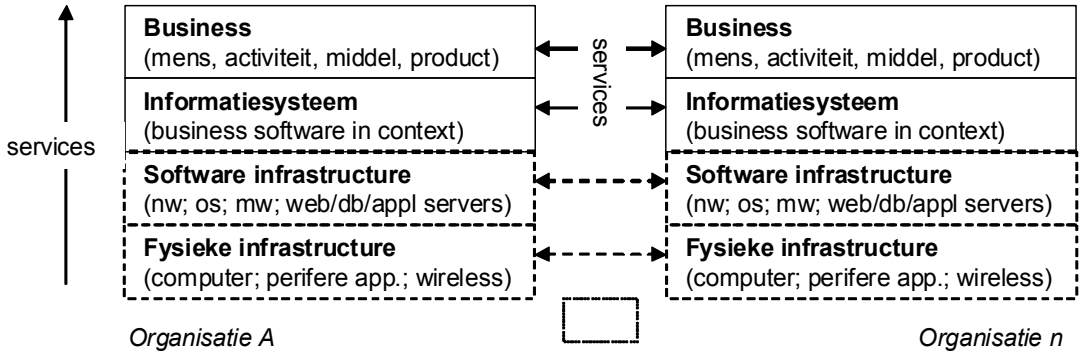


Figure 2: CNO (Collaborative Networked Organisation)

At the upper layer we find the cooperation, services and activities at the business level between people and organisations (e.g. giving insemination advice). The second layer is about business informationsystems (business software) that support the operations (e.g. milk equipment and farm management system). The third and fourth layer is infrastructure-oriented and contains the software and physical infrastructure that allows collaboration (networking, web servers, computers operating system, etc.).

The MOSES method focuses on the methodical design of electronic collaboration, the interaction between organisations and their informationsystems. The focus is thereby on the two upper layers of the four-layer architecture. We assume that a solution to interoperability is achieved based on a generic, omnipresent, technical infrastructure (the Internet with its standards and protocols). In Figure 2, we assume the software and physical infrastructure so if given.

3.2 The model-based approach

Our approach is based on the principle that all items relevant for the current interoperability issue are explicitly named and described in order to realize a better quality solution. The approach consists of 4 steps to get from describing the problem area to capturing the final technical solution:

1. Common Domain Model (CDM): in this step, the business domain of the considered problem area is captured and described. Concepts to be captured in the domain model

are for instance farm, cow, milk, food and so on. Also concepts in terms of events that take place in the business domain are captured, such as reporting milk production.

2. Common Information Model (CIM): in this step, a common information model for the domain is specified on the basis of the common domain model. Information elements in the this model comprise milk production data, milk yield, animal identification, food level and so on. This information has a more variable character than the concepts in the common domain model, but is still independent of a technical solution.

3. Common Message Model (CMM): in this step, on the basis of the CDM and the CIM, the messages for information exchange are defined. These messages refer to the information elements in the CIM (in the case the message is chosen as a paradigm solution direction). Think of messages like MilkingResult and LivestockList. This CMM also contains the accompanying dialogue processes in which these messages are used.

4. Translation to technical solution: Finally, a translation is made to a specific technological solution, in many cases in the form of XML messages or WSDL service specifications.

In the MOSES method, we make a formal distinction between specification of the common business processes and information exchange on the one hand and specification of the common solution to the interoperability issue on the other hand. The two pillars of the mindset when building the required models are:
1. The description of the business processes and information exchange is done independently of the description of the solution direction (business oriented domain).
2. Both descriptions are captured in models, which can be transformed to each other (model based).

In principle, the notation of the models to be developed can be done by any good modelling environment. For this document, we base our models on the UML diagram techniques.

Model	Type of content	Diagram
Common Domain Model	Actors and roles	UML Use Case diagram
	Domain objects	UML Class diagram
	Events	Eventlist
Common Information Model	Information objects and information elements	UML Class diagram
	Dictionary of terms	Table with all information elements
Common Message Model	Context diagram	UML Use case diagram
	Processes and activities	UML Activity diagram and/or UML Sequence diagram
	Message definitions	UML Class diagram
Technical Model	Message mapping	XML schemas, WSDL specs, mappings to existing common vocabularies

In the following chapters, we will further detail the Common Domain Model, Common Information Model and Common Message Model for the specific focus in the context of the insemination monitor. For the moment, the technical model and mapping towards technical implementations is out of scope.

4 Common Domain Model for milk production data exchange

This chapter describes the common domain model for the exchange of milk production data. Thereby, the scope is further narrowed down to the interaction between the milking equipment on the farm and the farm management system that keeps track of all the information around the herd of the farmer. This information system can be operated by (i) the farmer, (ii) the MRO or (iii) a third party. Within this interaction, the focus is on real-time milk production data. First, we describe the actors and their roles in the domain and the position of the various systems and interfaces between them. Second, we describe the main domain objects that are important for the scope of the interaction. Finally, we describe the specific events that trigger the interaction.

4.1 Actors and their roles in the domain

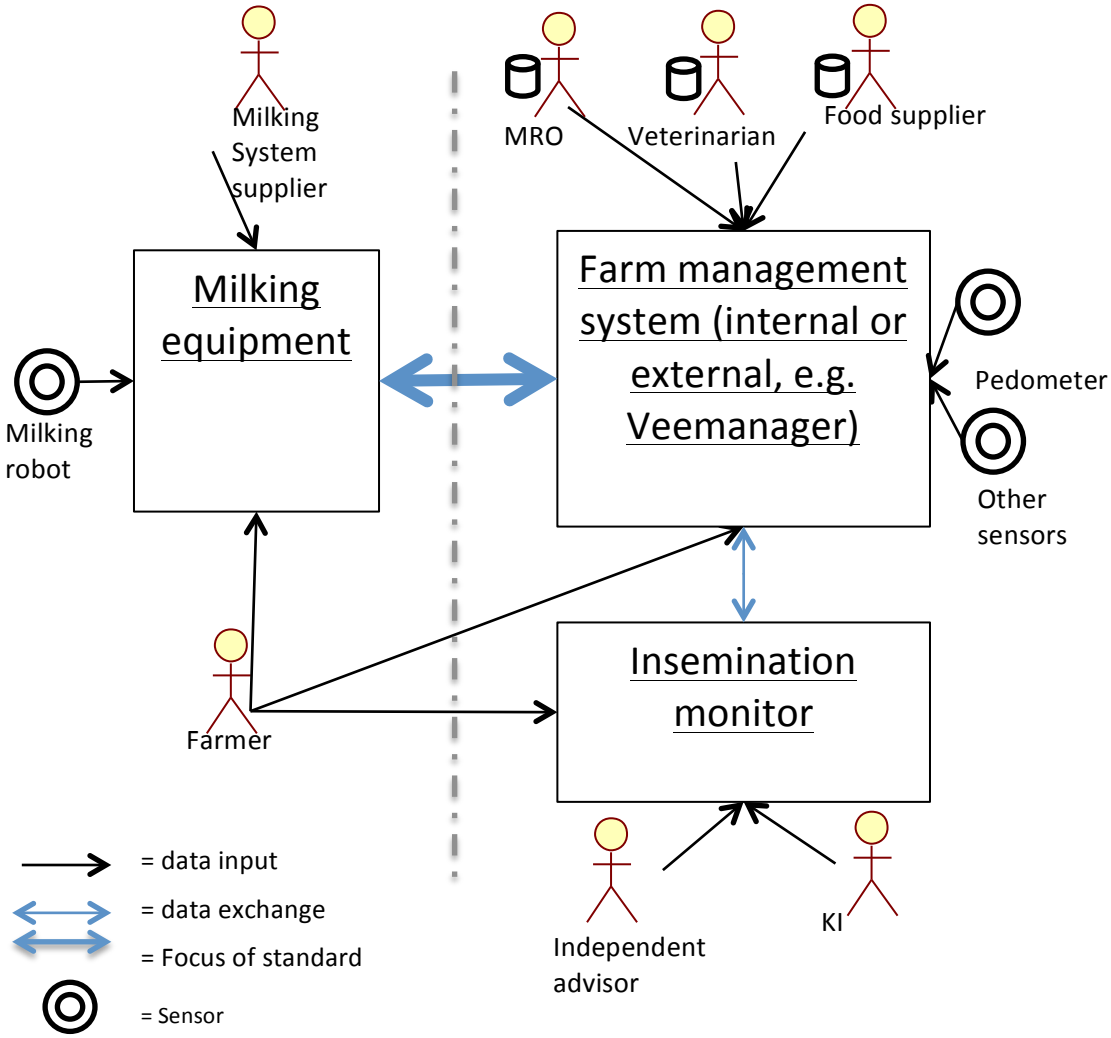


Figure 3: Common domain model

In section 2.2, we discussed the various stakeholders and their business cases or advantages with respect to the insemination monitor. In this section, we depict them in relation to the milking equipment, the farm management system and the insemination monitor. This is done because

we can then get a good view on the specific interface that we need for the interaction around milk production data. The focus of the standard is on the interface between the milking equipment and the farm management system. The interaction of the information system with other systems, e.g. systems of the MRO, the veterinarian, the food supplier or the milk company is out of scope for the moment.

We identified a number of actors with their roles in the domain (see domain model shown in Figure 3), however for the standard around milk production registration, only the first three actors in Table 1 are important.

Table 1 Actors relevant for the milk production registration standard

Actor	Definition
Farmer	Owner of a farm and a dairy herd, provides information about his cows.
Milking system supplier	Supplier of milking systems, such as milking robots, has to implement the standard.
Milk Recording Organisation (MRO)	Receives information about milking from farmers, analyses milk samples.
Veterinarian, Foods supplier, KI organisation, independent advisor	Out of the scope of our standard. Do not directly exchange information via the interface of the milking equipment and the farm management system.

4.2 Objects in the domain

The following model (Figure 4) shows the objects in the domain and their relations. The objects are described in Tabel 2.

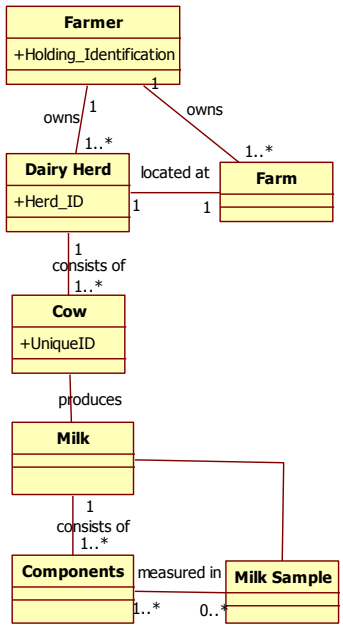


Figure 4: Object model

Tabel 2 Information objects in the domain

Object	Description
Farmer	Owner of a farm and a dairy herd, provides information about his cows.
Dairy herd	All female cows on a farm of a farmer.
Farm	Location where the farmer holds his cows.
Cow	Adult female cattle giving milk.
Milk	White liquid produced by adult female cows.
Characteristics	Includes components such as fat, lactose etc. found in the milk and other characteristics of the milk (e.g. conductivity).
Milk sample	Small quantity of milk analyzed to determine the milk components.

4.3 Events in the interaction between actors in the domain

We identify several events that take place around the milk production registration. An overview is provided in Tabel 3.

Tabel 3 List of events

Event	Description	Comments
animalDataList	The milking equipment signals the farm management system that it needs a herd list for a specific farm after which the farm management system composes an initial list of all active cows and sends it (with some attached details) to the milking equipment. The list is updated using transfer in and out messages.	The herd list contains all animals on the farm. The farmer can use parameters to determine which animals he wants to see. only the active cows. Cows that have been transferred can be updated on the list with transfer in and out messages.
registerMilking Results	The milking equipment sends a report with the milking results to the farm management system, operated by the farmer or the MRO.	This event can be triggered multiple times per day, in principle after each milking for each cow. The data is registered by the information system.
exchangeSample Information	The milking equipment sends the milk sample to the MRO and a message to the farm management system with the sampleID to be able to link the sample data with the individual cow. The MRO sends sample data about one individual milking or an aggregation of several milkings to the milking equipment for calibration and user presentation.	The milking equipment can be further fine-tuned based on the aggregated milking data of a cow.

In the figures below, the processes around the events are depicted.

4.3.1 Event: animalDataList

Figure 5 shows the event to exchange the herd list where either the milking equipment actively requests the herd list in order to be updated (pull, starting at the upper initial state), or the farm management system actively sends the herd list to the milking equipment (push, starting at the lower initial state). Figure 6 shows the event to update animal data on the herd list. We model both options i) the update is inserted at the milking equipment and send to the farm management system and ii) the update is inserted at the farm management system and send to the milking equipment. Both updates are send via TransferIn and TransferOut messages.

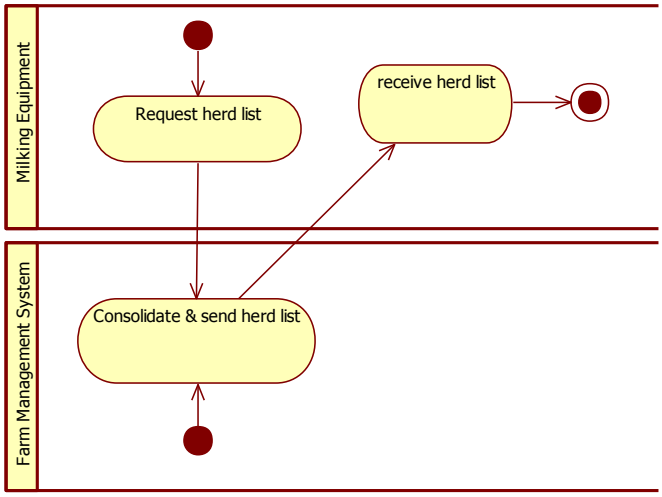


Figure 5: Event to exchange animal data

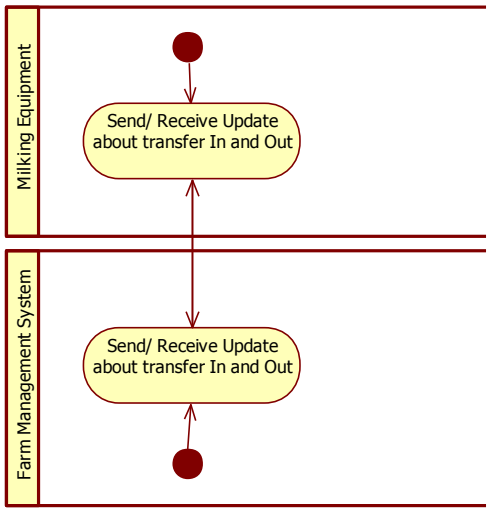


Figure 6: Event to update animal data

4.3.2 Event: registerMilkingResults

Figure 7 shows the event to register a set of milking results either for a single milking or a set of milkings per cow. We model the case where the milking equipment pushes the data towards the farm management system.

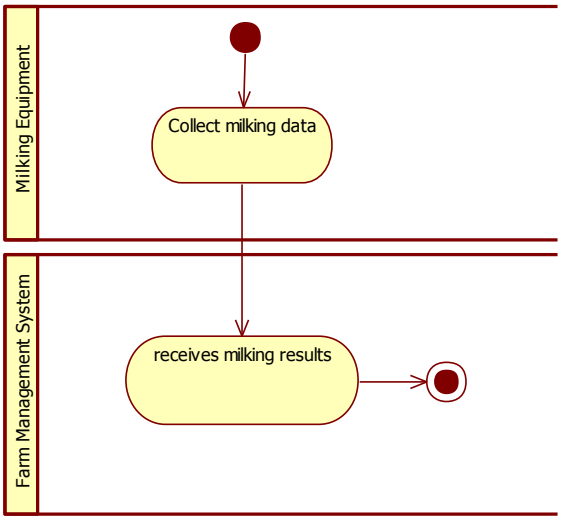


Figure 7: Event to register milking data

4.3.3 Event: exchangeSampleInformation

Figure 8 shows the event to register the sampleID in the farm management system and to collect information about the sample from the MRO to calibrate the milking equipment and to show to the user.

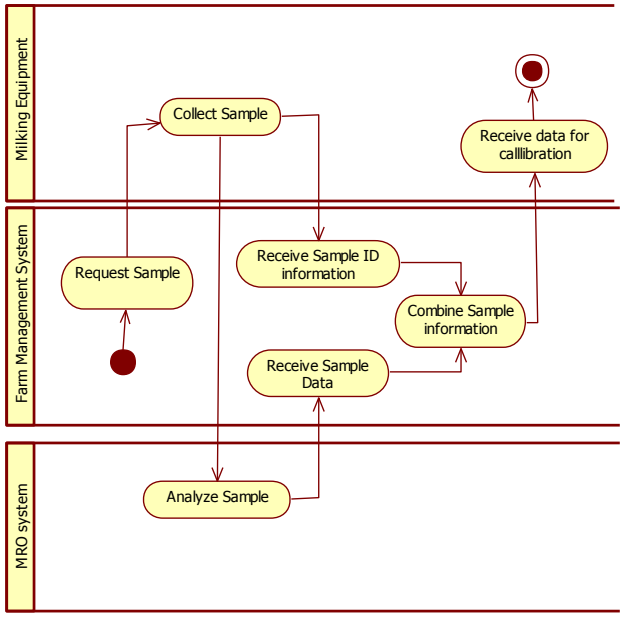


Figure 8: Event to register sampleID

5 Common Information Model for milk production data exchange

This chapter describes the common information model for the insemination monitor. Thereby, we first describe an overall common information model for all the information that is of interest for the insemination monitor. Afterwards we focus only on those information elements that are related to real-time milk production data exchanged between the milking equipment on the farm and the farm management system that keeps track of all the information around the herd of the farmer.

First, we describe all the information objects in an overall class diagram. This diagram also contains for instance a food object, although this information is of less relevance to the real-time milk production data. By doing this, we sketch how these information objects are positioned in the entire context of the insemination monitor. When the standard in this document has to be extended in the future, it is an easy step to extend the information model by detailing the information objects that are added to the scope.

Second, we describe the specific information elements in more detail. Thereby, the definition of each information element is given as precise as possible in order to define the semantics unambiguously. Also specific relations between information elements will be defined. Based on this chapter a separate vocabulary of the information elements in the common information model can be derived.

5.1 Information objects and their relation

We first present the common information model (Figure 9) and then zoom in on the scope of this project, namely those objects needed for the milk result registration.

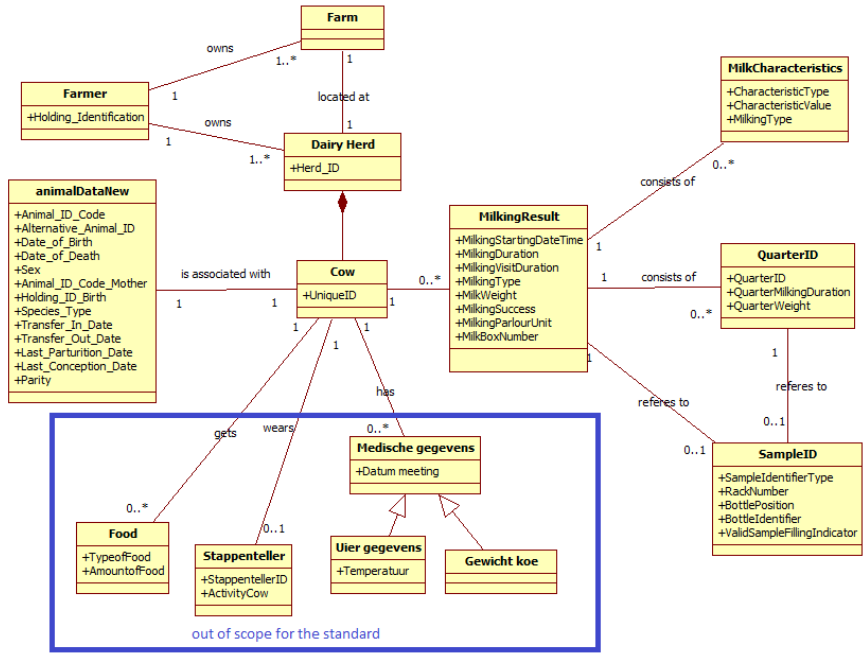


Figure 9: Common information model

Given the current scope of the standard we will first focus on the standardization of the actual milking results. We will complement it with some data about the animal itself and the farm. The picture of the overall information model shows that other data elements (blue box) can be added later on to the standard, e.g. information about the medical condition of a cow or about feeding.

5.2 Information elements

We introduce the following information elements and definitions.

5.2.1 Animal identification

For the animal identification we used input from the UN/CEFACT animal passport document as basis for the information elements and their definitions. We complemented these, if needed, with information elements described in other documents, e.g. the documents from ICAR. The animal identification information is important for the herd list.

Table 4 Information elements for herd list

Business term	**Rel.**	**Type**	**Description**	**Format**
Holding_Identification	Att	Identifier	Describe the holding or the premise for which the data is to be provided	String (12)
Herd_ID	Att	Identifier	Identifier of the herd the animal belongs to	String (15)
Animal_Identifier	Ass	Animal_ID	The animal identification code, if possible the lifetime number of the cow or the number appearing on the eartag that complies to the EU-wide identification coding system for animals	String (15)
Alternative_Animal_ID	Ass	Animal_ID	Alternative identification of the animal, e.g. “on farm numbers”	String (15)
Birth_Date_Time	Att	Date	Date birth of the animal	
Death_Date_Time	Att	Date	Date death of the animal	
Animal_Gender_Code	Att	Code	Sex of the animal M = male F = female	String (1)
Mother_Identifier	Att	Animal_ID	Identification code of the mother (e.g. Eartag number mother)	String (15)
RegisteredBirth_Location	Att	Identifier	Identifier of the farm where the animal is born or found	String (40)
Species_Type	Att	Code	Code for the species of the animal (Bovine, Sheep, Goat)	String (2)
Transfer_In_Date	Att	Date	Date the animal entered the current farm	
Transfer_Out_Date	Att	Date	Date the animal left the farm	
Last_Parturition_Date	Att	Date	Date the animal has given birth for the last time. (Last_Calving_Date)	
Last_Conception_Date	Att	Date	Date the animal got inseminated for the last time (Last_Insemination_Date)	
Parity	Att	Code	The number of times an animal has given birth.	String (2)

The type Animal_ID is defined as follows:

Businessterm	**Rel**
Code	Att
Value	Att

The code list for the Animal_ID is defined as follows:

Code	**Description**
1	Birthnumber
2	Respondernumber1
3	Respondernumber2
4	Cownumber
5	Brandnumber

The following important discussion points on the animal identification were raised and tackled by the working group:

- The Animal-ID: there is a European-wide coding system for animals, that have the form of a two character country-code followed by a 15-digit number. This number is provided by a national government body that is responsible for this. The name of the exact coding standard and the organisation that has set this standard has to be looked up.
- An “Alternative_Animal_ID” was added to the animal identification to account for other identification numbers the farmer might use. The animal should be identified with either the Animal-ID or the Alternative_Animal_ID or both.
- With respect to Date and Time we will use an international standard, e.g. ISO, UBL or UN/Cefact.
- We added Transfer-In-Date and Transfer-Out-Date to identify the animals that are currently on the farm. This is important information for the consolidation of the herd list. Thus, the MRO can maintain the history of animals in a herd and produce an actual herd list for a specific moment in time.

5.2.2 Milking result

Table 5 information elements actual milking result

Business term	**Rel.**	**Type**	**Description**	**Format**
MilkingStartingDateTime	Att	Date Time	Actual starting time of milking. Based on the definition of the milking equipment supplier, e.g. the moment the first teat liner is attached or the moment the first milking is streaming	General date time type
MilkingDuration	Att	Duration	Duration of the actual milking in seconds (ISO 3918 : duration on the milking machine	numeric(4)
MilkingVisitDuration	Att	Duration	Time between the moment the animal enters the milking equipment and the moment she exits the milking equipment (e.g. the box)	numeric(4)
MilkingType	Att	Code	Type of milking. See Code set below only for milk yield	string (1)
Code Set for Milking Type: 1: official milk control supplied by milk recording organisation 2 : measure of ICAR approved equipment				

3 : measure of not approved milking equipment				
MilkWeight	Att	Measure	Milk weight in Kg.	numeric(3.1)
MilkingSuccess	Att	Boolean	Indicator of the milking session validity as determined by the milking equipment supplier (Yes/ No)	Boolean
MilkingParlourUnit	Att	Identifier	Identification of the milking parlour unit	string(4)
MilkBoxNumber	Att	Identifier	Milking stand or box number	string(4)
QuarterMilking	Ass		The entity gives the results of the milking session for each quarter	
MilkCharacteristics	Ass		The entity provides information about the milk characteristics, including milk components	
SampleID	Ass		The entity provides the identification of the bottle which contain the sample	

The following important discussion points were raised and tackled by the working group:

- The starting time of a milking is hard to define unambiguously. Every milking equipment supplier has his own way of defining the starting time. We changed the descriptions of the information elements to make them more specific. The following picture might help to clarify the milking duration issue.

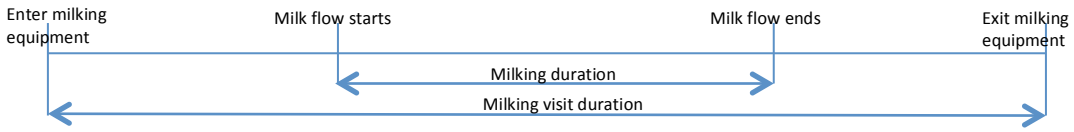


Figure 10: Milking duration

- The information element MilkingResultType was revised to MilkingType and the Code 9 (expected milking data) was deleted from the code list as it does not fit in this category.
- Constraint for MilkWeight: It is a required field if the “quarterIdentifier” is not filled out, as then the message is about the entire milking and thus the weight of the entire milking needs to be entered.
- The Valid-Milking-Indicator got changed to Milking-Success. The Code set got simplified to a Boolean Type, where a “Yes/ No” answer is possible. The criteria for milking success need to be determined by each milking equipment supplier individually.
- Conductivity and Flow Rate got combined with Milk-Components to a new entity called: Milk Characteristics
- MilkBoxNumber is added as an additional element to MilkingParlourUnit to allow for complete identification of the milking equipment.
- Several options exist to register the milking results for each quarter. One can either record the milking results in a seperate message/ entity for each quarter (then all the fields from the upper level message will be repeated here) (as currently proposed by ICAR) or specify in the TotalMilkingResult entity if the message is about one quarter or the total milking session (as currently done in NL). Then quarterIdentifier is added as attribute of the upper message (as shown above). The field is left empty if the data is about the total milking session. The latter option has been chosen.

Quarter Milking:

Table 6 Information elements quarter milking

Business term	**Rel.**	**Type**	**Description**	**Format**
QuarterIdentifier	Att	Code	Identification of the quarter for which the results apply: LF = left front RF = right front LR = left rear RR = right rear	String(2)
QuarterMilkingDuration	Att	Measure	Milking duration of the quarter	numeric(3)
QuarterMilkingWeight	Att	Measure	Milk weight of the quarter in Kg	numeric(3.1)

Milk Characteristics entity:

Table 7 Information elements milk characteristics

Business term	**Rel.**	**Type**	**Description**	**Format**
CharacteristicType	Att	Code	Type of milk characteristic or component measured	string(3)
CharacteristicValue	Att	Quantity	The measured value of the characteristic or component	string(12)
MilkingType	Att	Code	Type of milking. See Code set below only for milk yield	string (1)
Code Set for Milking Type: 1: official milk control supplied by milk recording organisation 2 : measure of ICAR approved equipment 3 : measure of not approved milking equipment				

- We identified two ways of defining components. We suggest to use the way ICAR defines MilkComponents, namely as a combination of ComponentType and ComponentValue. This way adjustments can easier be made in the future and the entity itself does not need to be adjusted. However, we have changed the name to CharacteristcType and Value in order to make it easily possible to extend the number of characteristics to be measured in the milk or the milking sample. The table below contains various possible characteristics to be measured in a sample.

Code Table of CharacteristicType

Code	Name	Description	Unit	Format
avgCnd	AverageConductivity	Average conductivity value of the milk in mS/cm at 25°C	Ms	numeric(2.1)
maxCnd	MaxConductivity	Maximum conductivity value of the milk in mS/cm at 25°C	Ms	numeric(2.1)
avgFlw	AverageFlowRate	Average flow rate for the individual milking in Kg/min	l/s	numeric(3.1)
maxFlw	MaxFlowRate	Max flow rate for the individual milking in Kg/min	l/s	numeric(3.1)
fat	Fat	Percentage fat at test day	%	numeric(4.2)
prtn	Protein	Percentage protein at test dat	%	numeric(4.2)
cllcnt	Cellcount	Somatic cell count	1000 cells/ml	numeric(4)

Apart from the specific characteristics, a set of metadata for the CharacteristicType is defined. See the table below.

Table 8 Metadata for Characteristic Type

Mult.	Business term	Rel.	Type	Description	Format
1	CharacteristicType	Att	Code	Type of the characteristic, e.g. component	string(3)
1	CharacteristicName	Att	Name	Name of the characteristic, e.g. component	string(20)
1	CharacteristicUnit	Att	Unit	Unit used for CharacteristicValue, e.g. %	String
1	CharacteristicPrec	Att	Precision	numerical precision used for CharacteristicValue e.g. 3.1	Numeric(2.2)
1	TypeCreationDate	Att	Date	Date for the creation of the type	
0..1	TypeRemovalDate	Att	Date	Date for the suppression of the type	

The metadata only needs to be entered once per CharacteristicType, e.g. fat. Afterwards the metadata entity only needs to be updated if something changes.

SampleID entity:

Table 9 Information elements to describe Sample ID

Business term	**Rel**	**Type**	**Description**	**Format**
MRO_ID	Att	Identifier	The identification of the MRO analysing the sample	String (?)
SampleIdentifierType	Att	Code	Type of bottle identifier. E.g. RFID, Barcode, Rack	String(1)
RackNumber	Att	Identifier	Number of the sample rack	string(6)
BottlePosition	Att	Identifier	Position of the bottle in the sample rack	String(3)
BottleIdentifier	Att	Identifier	Bottle identifiers read from barcode or RFID	String(20)
ValidSampleFillingIndicator	Att	Code	Indicator of valid sample filling compared with expected value. See code set.	String(1)
Aggregation	Att	Boolean	If the sample is an aggregation of several milkings of one cow, yes or no.	Boolean

For unique identification the Rack-Number and Bottle-Position are needed in the case the bottle is identified via the rack. If the bottle is identified by barcode or RFID only the BottleIdentifier is needed.

6 Common Message Model for milk production data exchange

We specify the following five messages:

1. initial herd list message,
2. transfer in message,
3. transfer out message,
4. milking results recording message and
5. sample message.

The unique key of each message is indicated in bold and explained after each table.

The described messages contain the fields for one animal, and we will only describe the multiplicity as if the message contains one animal.
In reality the messages can contain a zero or more animals. These multiplicities and relations will be described in the technical model.

6.1 Actual Initial Herd list message

The herd list message is needed to initially inform the milking equipment about the active animals on the farm that can be milked (active cows). Information about each animal (Animal identification) is send in the form of a herd list from the information system of the farmer to the milking equipment.

Actual initial herd list message:

Mult	Business term	Rel.
1	Holding_Identification	Att
1	**Herd_ID**	**Att**
1	Animal_ID	Att
0…n	Alternative_Animal_Id	Ass
0…1	Date_of_Birth	Att
1	Sex	Att
0…1	Transfer_In_Date	Att
0…1	Last_Partutition_Date	Att
0…1	Last_Conception_Date	Att
0…1	Parity	Att

Unique key: The combination of Holding_identification and Herd_ID form the unique key for the actual initial herd list message.
Constraints: Calving date and insemination date and parity should only be filled out when Sex is F (female).

6.2 Transfer In message

Once animals enter or leave the farm the herd list needs to be updated. The transfer in or out of an animal can be entered both at the milking equipment and the information system. The Transfer messages therefore can be send from the milking equipment to the information system and vice versa.

The transfer In message includes the same elements as the actual initial herd list message.

Mult	Business term	Rel.
1	Holding_Identification	Att
1	**Herd_ID**	**Att**
1	**Animal_ID**	**Att**
0…n	Alternative_Animal_Id	Att
0…1	Date_of_Birth	Att
1	Sex	Att
0…1	**Transfer_In_Date**	**Att**
0…1	Last_Calving_Date	Att
0…1	Last_Insemination_Date	Att
0…1	Parity	Att

Unique key: The combination of Herd_ID, Animal_ID and Transfer_In_Date form the unique key for the transfer in message.

6.3 Transfer Out message

Mult	Business term	Rel.
1	Holding_Identification	Att
1	**Herd_ID**	**Att**
1	**Animal_ID**	**Att**
0…n	Alternative_Animal_Id	Att
1	**Transfer_Out_Date**	**Att**

Unique key: The combination of Herd_ID, Animal_ID and Transfer_ut_Date form the unique key for the transfer out message.

6.4 Milking results recording message

The milking equipment sends a message about the milking results to the information system with information about the total milking. This includes all of the elements defined in Chapter 5, in the entity:

Milking-Results message

Mult	Business term	Rel.
1	Holding_Identification	Att
1	**Herd_ID**	**Att**
1	**Animal_ID**	**Att**
0…n	Alternative_Animal_Id	Att

1	**MilkingStartingDateTime**	**Att**
0…1	MilkingDuration	Att
0…1	MilkingVisitDuration	Att
1	MilkingType	Att
1	MilkWeight	Att
1	MilkingSuccess	Att
0…1	MilkingParlourUnit	Att
0…1	MilkBoxNumber	Att
0…4	QuarterMilking	Ass
0…n	MilkCharacteristics	Ass
0…1	SampleID	Ass

Unique key: The combination of Herd_ID + Animal_ID + MilkingStartingDateTime is the unique key for this message. If Animal_ID is a worldwide unique number then one only needs Animal_ID and MilkingStartingDateTime to have a unique key.

We define the following three subsets of the milking results message:

Subset of message for quarter milking:

Mult	**Business term**	**Rel.**
1	**QuarterIdentifier**	**Att**
0…1	QuarterMilkingDuration	Att
0…1	QuarterMilkingWeight	Att
0…n	MilkCharacteristics	Ass
0…1	SampleID	Ass

Unique key: The combination of the unique key from the milking result message with the quarter identifier.

Subset of message for milk characteristics:

Mult	**Business term**	**Rel.**
1	**CharacteristicType**	**Att**
1	CharacteristicValue	Att
1	MilkingType	Att

Unique key: The combination of the unique key from the milking result message with the charactersiticType.

Subset of message for sample ID:

Mult	**Business term**	**Rel.**
1	SampleIdentifierType	Att
0...1	**RackNumber**	**Att**
0..1	**BottlePosition**	**Att**
0..1	**BottleIdentifier**	**Att**
0..1	ValidSampleFillingIndicator	Att

Unique key: The combination of the unique key from the milking result message with either the RackNumber + Bottle Position or the BottleIdentifier.

6.5 Sample message

This message is send from the information systems (from MRO) to the milking equipment with data about the sample. This information is used by the milking equipment for calibration and presentation to the users. This information can be on one individual milking or aggregated data.

Calibration message about sample:

Mult	Business term	Rel.
1	MRO_ID	Att
1	Holding_Identification	Att
1	**Herd_ID**	**Att**
1	**Animal_ID**	**Ass**
0…n	Alternative_Animal_Id	Ass
1	**MilkingStartingDateTime**	**Att**
0…n	SampleID	Ass
0…n	MilkingCharacteristics	Ass
0…4	QuarterMilking	Ass
1	Aggregation	Att

Unique key:
The combination of Herd_ID, Animal_ID and MilkingStartingDateTime form the unique key for the sample message.

7 Technical Model

7.1 Use Cases

This chapter takes all previous chapters and combines them into a technical model. Based on the events as described in 4.3, use-cases are created to identify the communication messages needed to enable the events. Each use case involves set of messages, which will consist of a request and a response message. These request and response messages will be based on the information defined in the common message model chapter 6. However, here these messages will be translated to a technical implementation, and can result in multiple xml messages per defined message from chapter 6, or be combined into one message.

7.1.1 Event: animalDataList

Used messages

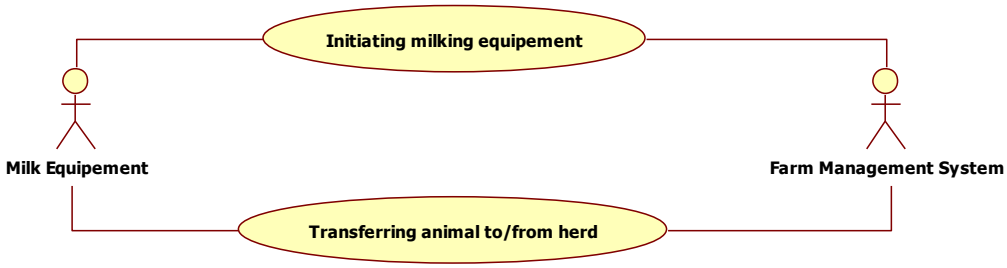


Figure 11: Use cases for event animalDataList

The AnimalDataList event has 2 use-cases

- use-case: Initiating milking equipment-When milk equipment (ME) is placed at a location, it has no information about the herd it is going to be milking. To get this information a request, containing the location of the ME, will be send to the Farm Management system. The FMS (Farm Management system) will collect all know data for that location and send it back in a response message

- use-case: Transferring animal to/from herd
  When an animal changes herds, the ME or the FMS needs to be informed. The initiative for this use case can be the ME of the FMS. If an animal enters a herd or leaves a herd (for example: in case of death), without the FMS being informed, the ME can send a transfer message to the FMS to inform about the change in the herd. The FMS can also inform a ME about new animals in his herd or animals that left the herd.

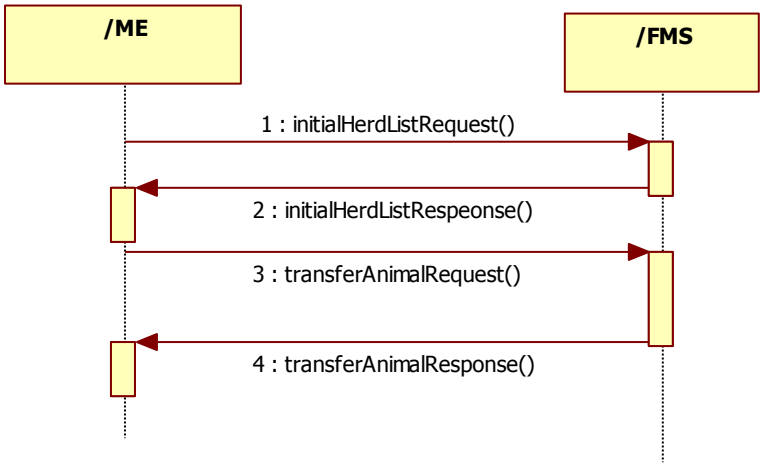


Figure 12: Sequence diagram for the use cases animalDataList

For this this event we have defined 4 messages that will enable the three use-cases.

XML message	Used for	contains	Based on Common Message Model
initialHerdListRequest	Requesting an herd list	The holding the list request is from	
initialHerdListResponse	Send a complete herd list	The herd list for the requested holding	initial herd list message
transferAnimalRequest	Transfer an animal to or out of a herd	Animal information from either the ME of the FMS with an indicator for transferring in or out	transfer in message transfer out message
transferAnimalResponse	Acknowledge the transfer	A standard response	

7.1.2 Event: registerMilkingResults

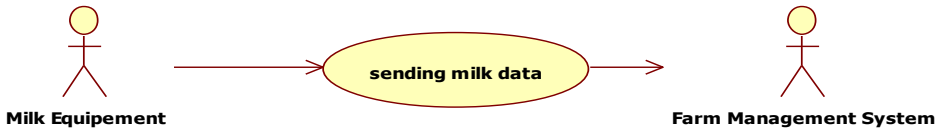


Figure 13: Use cases from event registerMilkingResults

- use-case: Sending milk data
  After an animal/herd has been milked the milking data will be send to the FMS. The FMS then store this information. The sending of the information can be done periodically or after every milking session. The data from these milking sessions can be used to manage and plan inseminations or monitor milk production.

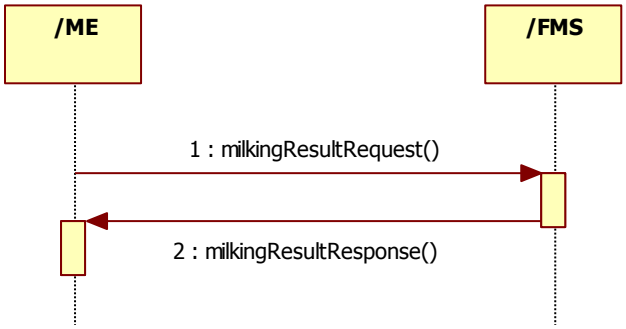


Figure 14: Sequence diagram for the use cases: registerMilkingResults

2 messages will be defined to accommodate this use case.

XML message	Used for	contains	Based on Common Message Model message
milkingResultRequest	Sending the milking results	The milking results	Milking results recording message
milkingResultResponse	Acknowledge the milking results	The standard response	

7.1.3 Event: exchangeSampleInformation

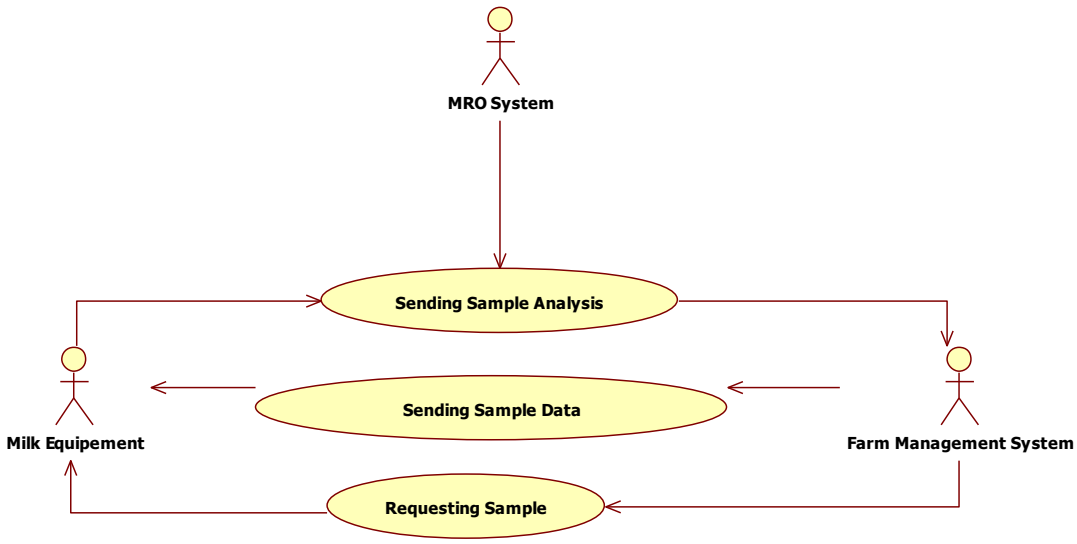


Figure 15: Use cases from event exchangeSampleInformation

- Use Case: Requesting sample data
  Once in a while the FMS will request a sample to be taken

- Use Case: Sending Sample Analysis
  After the lab or the ME have analyzed the sample, they will send the results to the FMS. The ME will combine the results with the milk results, the MRO system however will send only the results.

- Use Case: Sending Sample Data
  If the FMS receives an analysis from the RMO system it will find the milking sample the analysis belongs to and send this sample data to the ME.

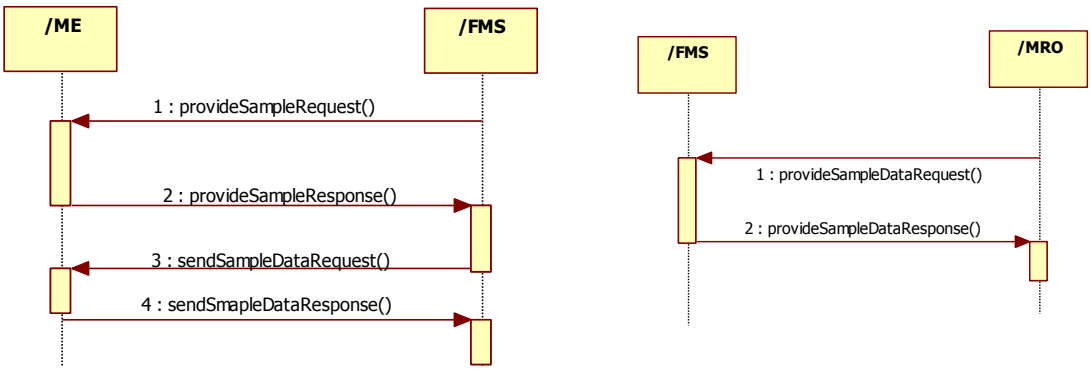


Figure 16: Sequence diagrams for the use cases: exchangeSampleInformation

For this use cases 4 new messages will be defined. However the sample data message information is also part of the milking result message. This means the data message will be send from the RMO to the FMS, which then will send the message to the ME. The ME itself will not send a sample data message to the FMS.

XML message	**Used for**	**contains**	**Based on Common Message Model message**
provideSampleRequest	Asking for a sample to be taken	The animal information the sample has to be taken from	
provideSampleResponse	Acknowledge request	The standard response	
sendSampleDataRequest	Sending sample results	The result of the sample analysis	Sample message
sendSampleDataResponse	Acknowledge data request	The standard response	

7.2 Messages

In total 10 new messages, consisting of 5 request and 5 response messages, will be defined to support the communication between the ME, the FMS and the MRO.

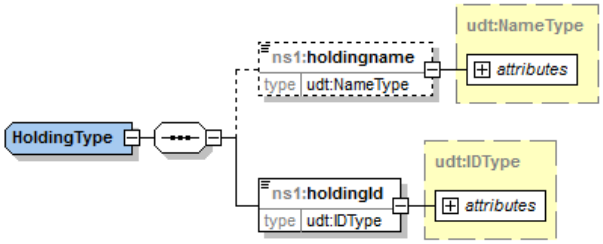
All messages will make use of the UN/CEFACT unqualified data types defined in:

- http://www.unece.org/uncefact/data/standard/UnqualifiedDataType_13p0.xsd

7.2.1 ComplexTypes

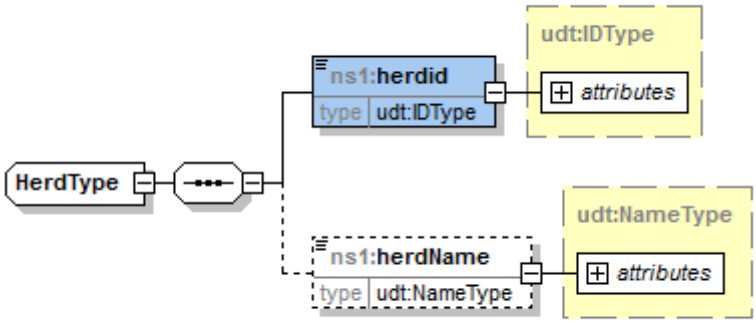
A complex type element is an XML element that contains other elements and/or attributes. (W3schools.com) Complextypes are used here for reusability and to make the xsd less complex and more understandable.

Holdingtype



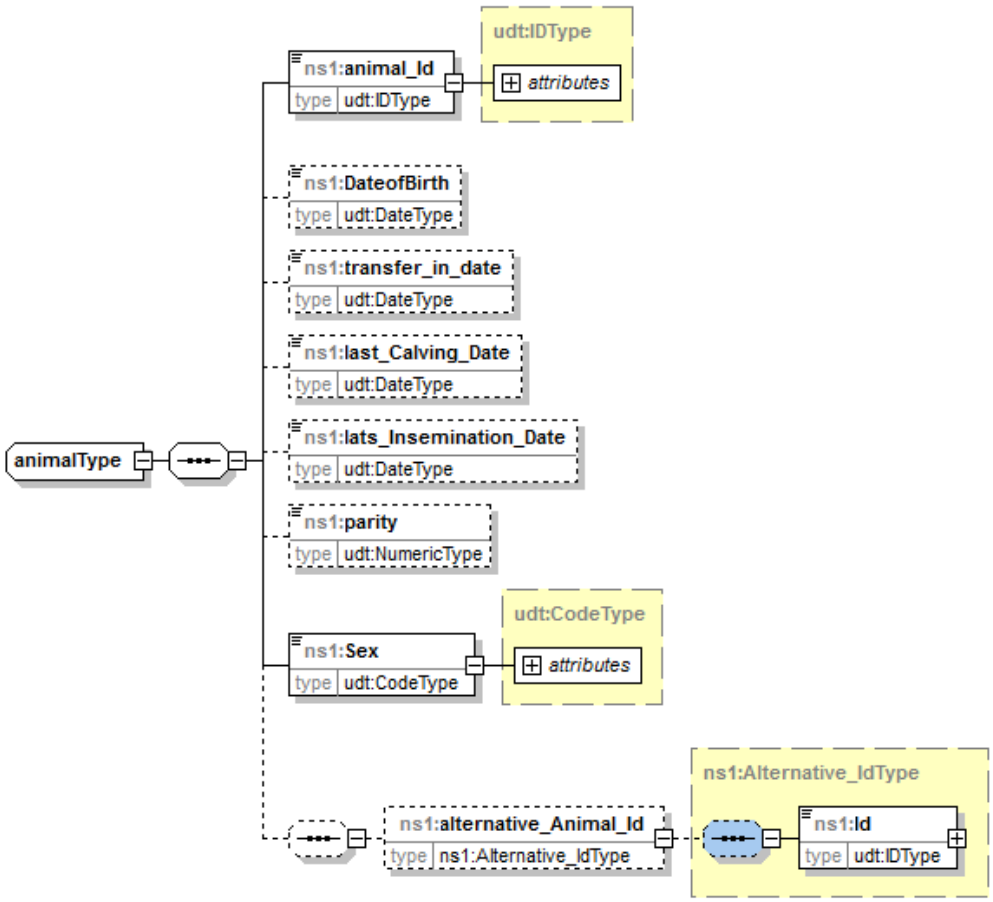
The holdingtype defines a holding and contains the name of the holding and the unique identifier for the holding.

Herdtype



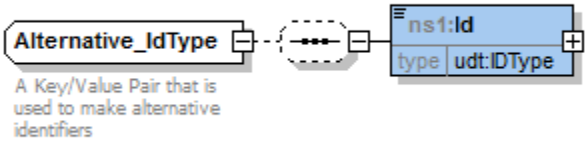
The Herdtype describes the herd an animal is located in. It defines the name of the herd and the unique identifier of the herd.

animalType



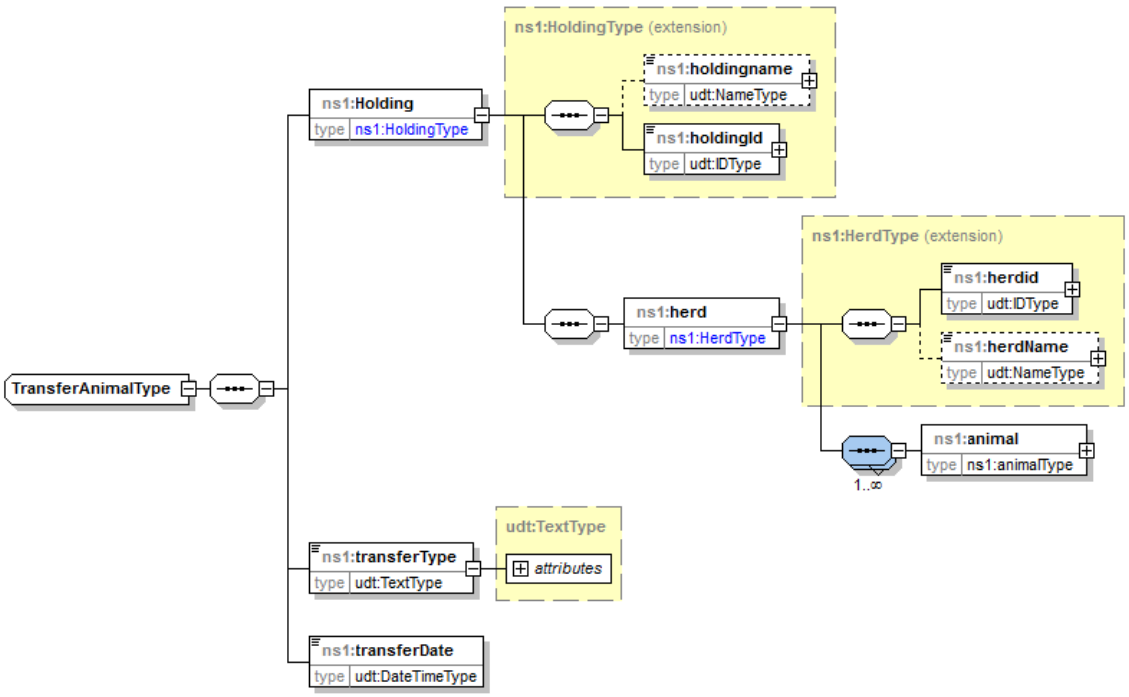
The animaltype defines an Animal. Outside from the unique identifier used by the system, it also contains a list of alternative identifiers.

Alternative_IdType



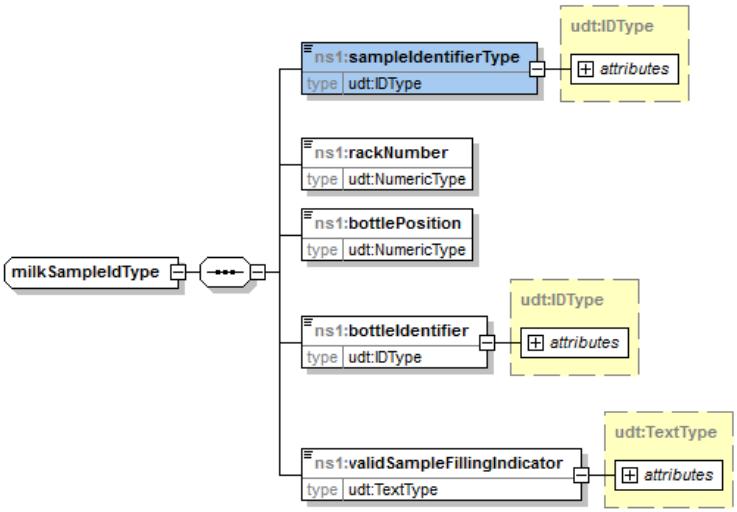
The alternative_IdType can hold the unique identifiers other than the identifier used by the system. It uses the IDtype defined in the UN/CEFACT XSD. This IDtype defines the scheme the ID is described and the agency the identifier has been given out by.

transferAnimalType



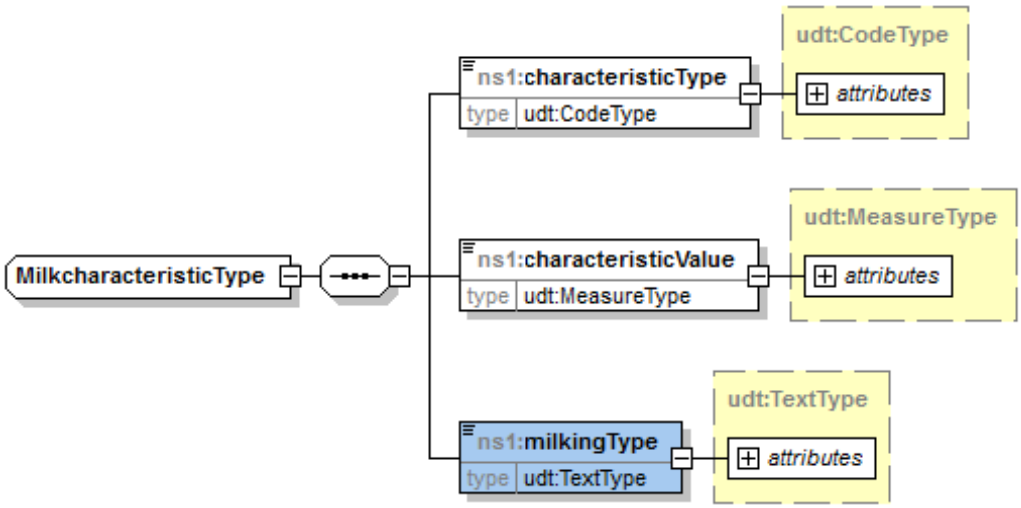
The animalTransferType is used to describe to transfer information that is needed to transfer an animal in or out of a herd. It used the HolindgType, HerdType and the AnimalType to combine the information, and uniquely identify the herd and the animal. The transferType is used to distinguish a transfer to or from a herd.

milkSampleIdType



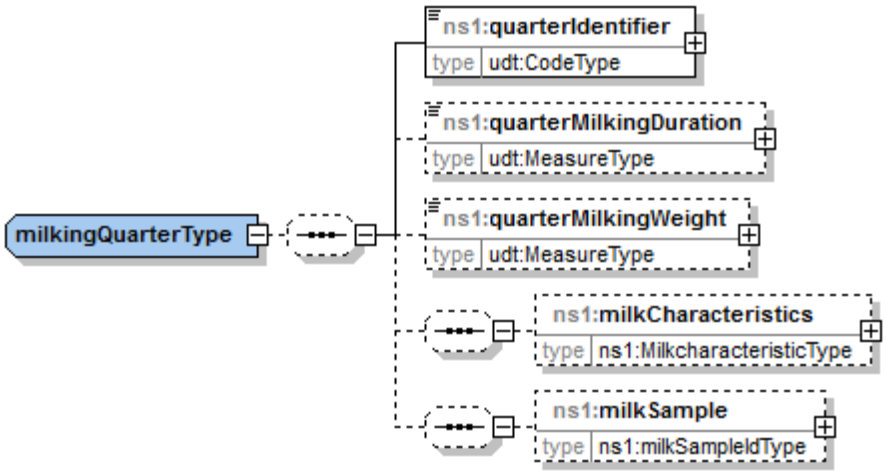
The milkSampleType describes the information the ME sends to the FMS about a sample that has been taken from a milking session. It contains the identifiers needed to identify the sample.

milkCharateristicsType



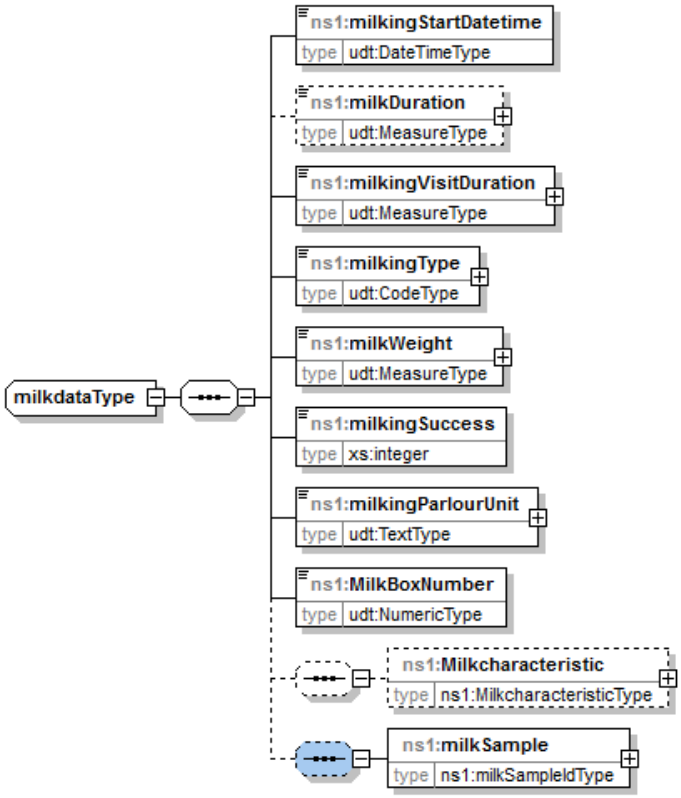
A milkingCharacteristicsType is used to store information about a characteristic found in a milking session of a milking sample.

milkingQuaterType



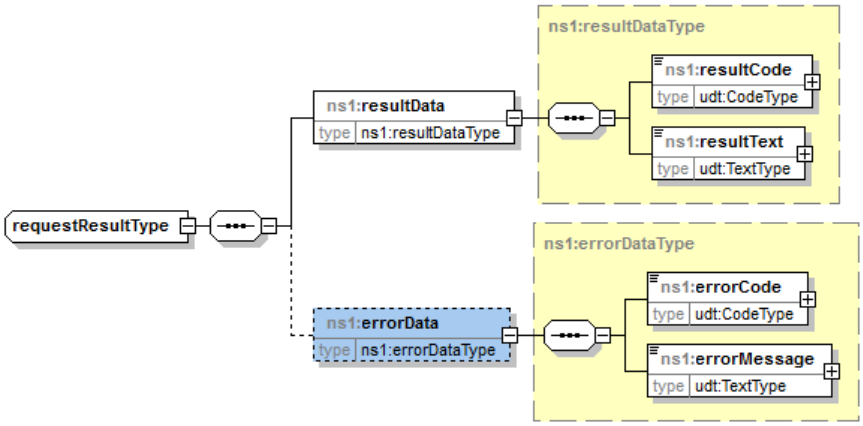
Milking is done in 4 quarters. The milkingQuarterType defines each quarter.

milkDataType



The milkDataType describes the result s for as milking sessions.

requestResultType, errorDataType and resultDataType



The requestResultType is used in the standard response message described in chapter 7.2.2

7.2.2 Standard Response messages

In order to have a good communication between parties every request will need a response. Therefore we defined a standard response message. A standard response message will always contain a requestResult.

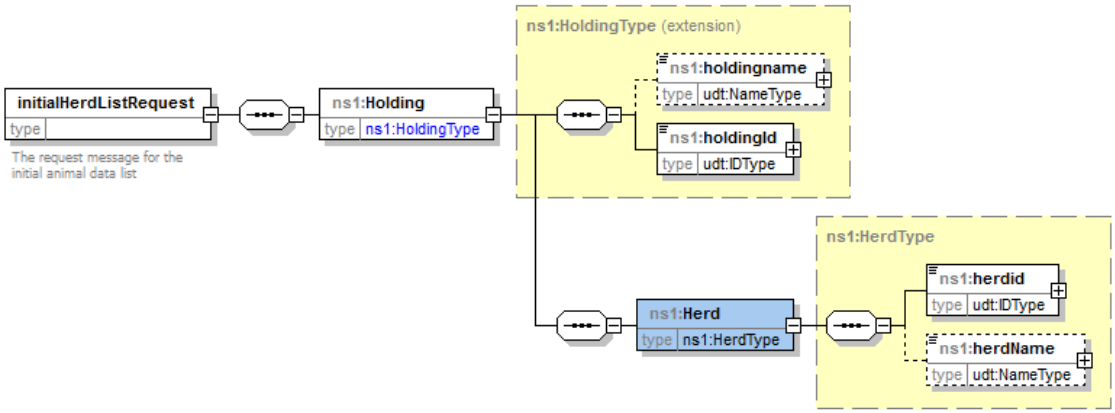
This requestResult consist of 2 parts. The first part will be the resultData. This code will be 0 if the request was received and handled correctly. If the request was not received in good order the code will be -1. If the request was not processed correctly this code will be -2. The second part is the errorData. If the result is not equal to 0, the erroCode will contain the reason why the request was not received or processed correctly.

Some requests require a response with more content. These messages will be described separately in the next chapters.

7.2.3 initialHerdListRequest and initialHerdListResponse

The animalDataListRequest message will request the initial list from the FMS for fill the initial database at the ME.

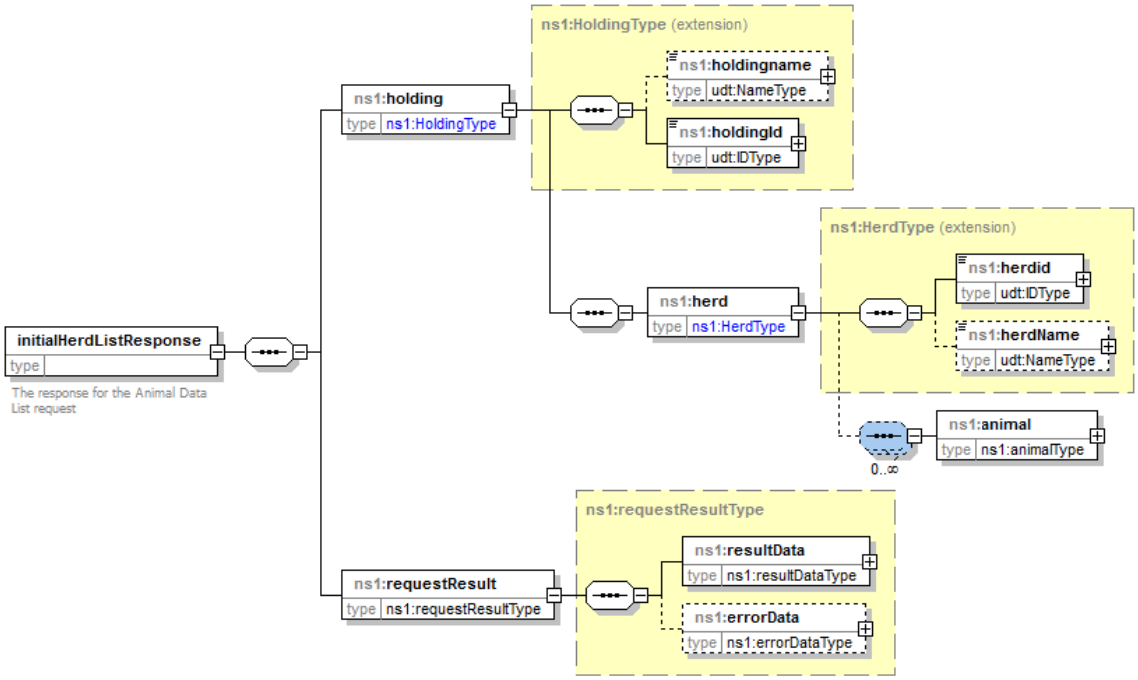
The request will only contain the holding. This identifies the place the ME is located
Based on this holding the FMS can create a herd/animal list.



Example :

```
<ns1:initialHerdListRequest xsi:schemaLocation="SDF_MilkingHerdMessageType:1
SDFMilkingMessage.xsd" xmlns:ns1="SDF_MilkingHerdMessageType:1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <ns1:Holding>
      <ns1:holdingname>De blije Koe</ns1:holdingId>
      <ns1:holdingId>2034</ns1:holdingId>
   </ns1:Holding>
</ns1:initialHerdListRequest>
```

As a response to the animalDataListRequest the FMS will send the complete herd list to the ME



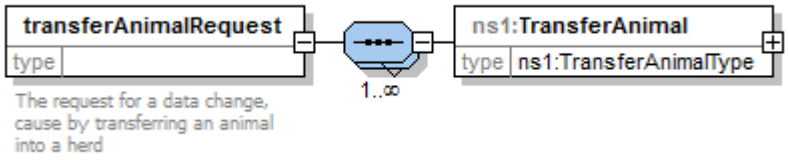
The response will contain the holding where the herd is located. This way the receiver can confirm the right herd is being send.

Example:

```
<ns1:initialHerdListResponse xsi:schemaLocation="SDF_MilkingHerdMessageType:1
SDFMilkingMessage.xsd" xmlns:ns1="SDF_MilkingHerdMessageType:1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <ns1:holding>
      <ns1:holdingname languageID="en-us">De blije Koe </ns1:holdingname>
      <ns1:holdingId schemeID="token" schemeName="String" schemeAgencyID="201"
                    schemeAgencyName="String" schemeVersionID="token"
                    schemeDataURI="http://www.altova.com"
                    schemeURI="http://www.altova.com">2034</ns1:holdingId>
      <ns1:herd>
         <ns1:herdid schemeID="token" schemeName="String" schemeAgencyID="33"
                     schemeAgencyName="String" schemeVersionID="token"
                     schemeDataURI="http://www.altova.com"
                     schemeURI="http://www.altova.com">A3</ns1:herdid>
         <ns1:herdName languageID="en-us">A-Team</ns1:herdName>
         <ns1:animal>
            <ns1:animal_Id schemeID="token" schemeName="String" schemeAgencyID="61"
                           schemeAgencyName="String" schemeVersionID="token"
                           schemeDataURI="http://www.altova.com"
                           schemeURI="http://www.altova.com">34563</ns1:animal_Id>
            <ns1:DateofBirth>1972-06-23</ns1:DateofBirth>
            <ns1:transfer_in_date>1978-08-13</ns1:transfer_in_date>
            <ns1:last_Calving_Date>1979-03-04</ns1:last_Calving_Date>
            <ns1:lats_Insemination_Date>1983-08-04</ns1:lats_Insemination_Date>
            <ns1:parity>2</ns1:parity>
            <ns1:Sex listID="token" listAgencyID="142" listAgencyName="String"
                     listName="String" listVersionID="token" name="String" languageID="en-us"
                     listURI="http://www.altova.com"
                     listSchemeURI="http://www.altova.com">F</ns1:Sex>
```

```
            <ns1:alternative_Animal_Id>
               <ns1:Id schemeID="token" schemeName="String" schemeAgencyID="309"
                       schemeAgencyName="String" schemeVersionID="token"
                       schemeDataURI="http://www.altova.com"
                       schemeURI="http://www.altova.com">EAR4563</ns1:Id>
            </ns1:alternative_Animal_Id>
         </ns1:animal>
      </ns1:herd>
   </ns1:holding>
   <ns1:requestResult>
      <ns1:resultData>
         <ns1:resultCode listID="token" listAgencyID="28" listAgencyName="String"
                         listName="String" listVersionID="token" name="String"
                         languageID="en-us" listURI="http://www.altova.com"
                         listSchemeURI="http://www.altova.com">0</ns1:resultCode>
         <ns1:resultText languageID="en-us">Success</ns1:resultText>
      </ns1:resultData>
      <ns1:errorData>
      </ns1:errorData>
   </ns1:requestResult>
</ns1:initialHerdListResponse>
```

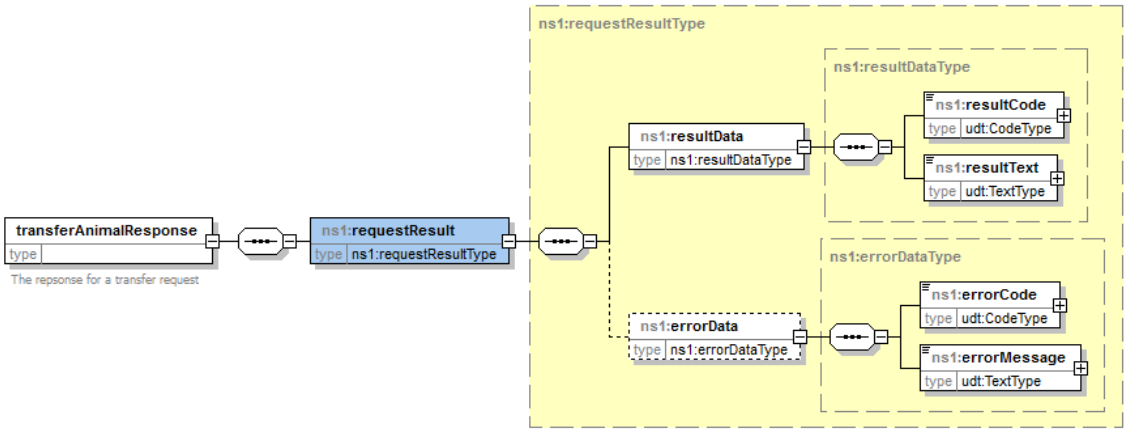
7.2.4 transferAnimalRequest and transferAnimalResponse



Example:

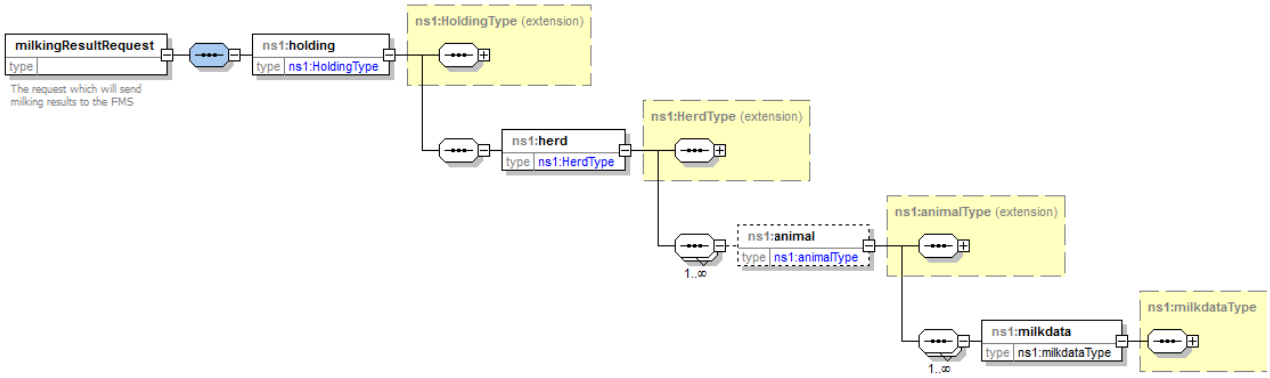
```
<ns1:transferAnimalRequest xsi:schemaLocation="SDF_MilkingHerdMessageType:1
SDFMilkingMessage.xsd" xmlns:ns1="SDF_MilkingHerdMessageType:1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <ns1:TransferAnimal>
      <ns1:holding>
         <ns1:holdingname languageID="en-us">De blije Koe </ns1:holdingname>
         <ns1:holdingId schemeID="token" schemeName="String" schemeAgencyID="201"
                       schemeAgencyName="String" schemeVersionID="token"
                       schemeDataURI="http://www.altova.com"
                       schemeURI="http://www.altova.com">2034</ns1:holdingId>
         <ns1:herd>
         <ns1:herdid schemeID="token" schemeName="String" schemeAgencyID="33"
                       schemeAgencyName="String" schemeVersionID="token"
                       schemeDataURI="http://www.altova.com"
                       schemeURI="http://www.altova.com">A3</ns1:herdid>
         <ns1:herdName languageID="en-us">A-Team</ns1:herdName>
            <ns1:animal>
               <ns1:animal_Id schemeID="token" schemeName="String" schemeAgencyID="61"
                             schemeAgencyName="String" schemeVersionID="token"
                             schemeDataURI="http://www.altova.com"
                             schemeURI="http://www.altova.com">34563</ns1:animal_Id>
               <ns1:alternative_Animal_Id>
                  <ns1:Id schemeID="token" schemeName="String" schemeAgencyID="309"
```

```
                    schemeAgencyName="String" schemeVersionID="token"
                    schemeDataURI="http://www.altova.com"
                    schemeURI="http://www.altova.com">EAR4563</ns1:Id>
                </ns1:alternative_Animal_Id>
            </ns1:animal>
        </ns1:herd>
    </ns1:Holding>
    <ns1:transferType languageID="en-us">In</ns1:transferType>
    <ns1:transferDate>2001-12-17T09:30:47Z</ns1:transferDate>
  </ns1:TransferAnimal>
</ns1:transferAnimalRequest>
```




```
<ns1:transferAnimalResponse xsi:schemaLocation="SDF_MilkingHerdMessageType:1
    SDFMilkingMessage.xsd" xmlns:ns1="SDF_MilkingHerdMessageType:1"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <ns1:requestResult>
    <ns1:resultData>
      <ns1:resultCode listID="token" listAgencyID="25" listAgencyName="String"
                      listName="String" listVersionID="token" name="String"
                      languageID="en-us" listURI="http://www.altova.com"
                      listSchemeURI="http://www.altova.com">-1</ns1:resultCode>
      <ns1:resultText languageID="en-us">Error</ns1:resultText>
    </ns1:resultData>
    <ns1:errorData>
      <ns1:errorCode listID="token" listAgencyID="248" listAgencyName="String"
                     listName="String" listVersionID="token" name="String"
                     languageID="en-us" listURI="http://www.altova.com"
                     listSchemeURI="http://www.altova.com">3</ns1:errorCode>
      <ns1:errorMessage languageID="en-us">Animal not found</ns1:errorMessage>
    </ns1:errorData>
  </ns1:requestResult>
</ns1:transferAnimalResponse>
```

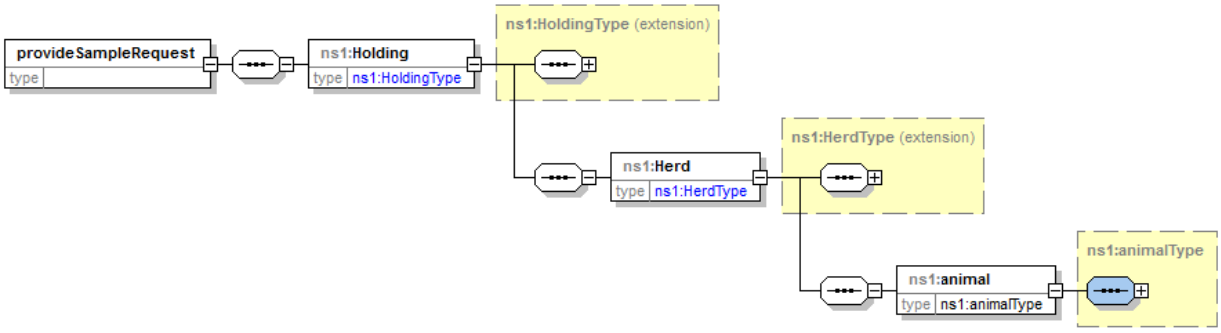
7.2.5 milkingResultRequest and milkingResultResponse



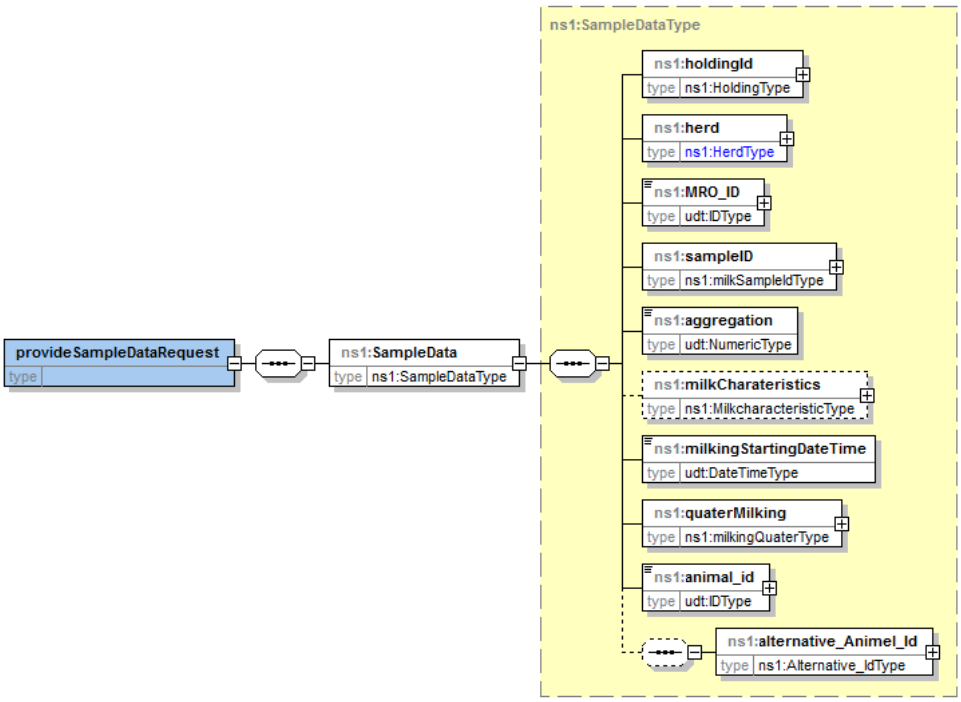

```
<ns1:milkingResultRequest xsi:schemaLocation="SDF_MilkingHerdMessageType:1
                                SDFMilkingMessage.xsd"
                                mlns:ns1="SDF_MilkingHerdMessageType:1"
                                xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <ns1:holding>
      …
      <ns1:herd>
         …
         <ns1:animal>
            …
            <ns1:milkdata>
               <ns1:milkingStartDatetime>2001-12-17T09:30:47Z</ns1:milkingStartDatetime>
               <ns1:milkDuration unitCode="5B">0.0</ns1:milkDuration>
               <ns1:milkingVisitDuration unitCode="74">0.0</ns1:milkingVisitDuration>
               <ns1:milkingType listID="token" listAgencyID="50" listAgencyName="String"
listName="String" listVersionID="token" name="String" languageID="en-us"
listURI="http://www.altova.com"
listSchemeURI="http://www.altova.com">token</ns1:milkingType>
               <ns1:milkWeight unitCode="G32">0.0</ns1:milkWeight>
               <ns1:milkingSuccess>0</ns1:milkingSuccess>
               <ns1:milkingParlourUnit languageID="en-us">String</ns1:milkingParlourUnit>
               <ns1:MilkBoxNumber>0.0</ns1:MilkBoxNumber>
               <ns1:Milkcharacteristic>
                  <ns1:characteristicType listID="token" listAgencyID="200"
listAgencyName="String" listName="String" listVersionID="token" name="String"
languageID="en-us" listURI="http://www.altova.com"
listSchemeURI="http://www.altova.com">token</ns1:characteristicType>
                  <ns1:characteristicValue unitCode="E15">0.0</ns1:characteristicValue>
                  <ns1:milkingType languageID="en-us">String</ns1:milkingType>
               </ns1:Milkcharacteristic>
               <ns1:milkSample>
                  <ns1:sampleIdentifierType schemeID="token" schemeName="String"
schemeAgencyID="188" schemeAgencyName="String" schemeVersionID="token"
schemeDataURI="http://www.altova.com"
schemeURI="http://www.altova.com">token</ns1:sampleIdentifierType>
                  <ns1:rackNumber>0.0</ns1:rackNumber>
                  <ns1:bottlePosition>0.0</ns1:bottlePosition>
```

```
                <ns1:bottleIdentifier schemeID="token" schemeName="String"
schemeAgencyID="22" schemeAgencyName="String" schemeVersionID="token"
schemeDataURI="http://www.altova.com"
schemeURI="http://www.altova.com">token</ns1:bottleIdentifier>
                <ns1:validSampleFillingIndicator languageID="en-
us">String</ns1:validSampleFillingIndicator>
              </ns1:milkSample>
            </ns1:milkdata>
          </ns1:animal>
        </ns1:herd>
      </ns1:holding>
</ns1:milkingResultRequest>
```

7.2.6 provideSampleRequest and provideSampleResponse



7.2.7 sendSampleDataRequest and sendSampleDataResponse



Example:

```
<ns1:provideSampleDataRequest xsi:schemaLocation="SDF_MilkingHerdMessageType:1
SDFMilkingMessage.xsd" xmlns:ns1="SDF_MilkingHerdMessageType:1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <ns1:SampleData>
      <ns1:holdingId>
      …
      <ns1:herd>
         …
         <ns1:animal>
            …
         </ns1:animal>
      </ns1:herd>
      <ns1:MRO_ID schemeID="token" schemeName="String" schemeAgencyID="50"
                  schemeAgencyName="String" schemeVersionID="token"
                  schemeDataURI="http://www.altova.com"
                  schemeURI="http://www.altova.com">token</ns1:MRO_ID>
      <ns1:sampleID>
         <ns1:sampleIdentifierType schemeID="token" schemeName="String"
                                   schemeAgencyID="81" schemeAgencyName="String"
                                   schemeVersionID="token"
                                   schemeDataURI="http://www.altova.com"
                           schemeURI="http://www.altova.com">token</ns1:sampleIdentifierType>
         <ns1:rackNumber>0.0</ns1:rackNumber>
         <ns1:bottlePosition>0.0</ns1:bottlePosition>
         <ns1:bottleIdentifier schemeID="token" schemeName="String" schemeAgencyID="103"
                               schemeAgencyName="String" schemeVersionID="token"
                               schemeDataURI="http://www.altova.com"
                               schemeURI="http://www.altova.com">token</ns1:bottleIdentifier>
         <ns1:validSampleFillingIndicator languageID="en-us">
         String</ns1:validSampleFillingIndicator>
      </ns1:sampleID>
      <ns1:aggregation>0.0</ns1:aggregation>
      <ns1:milkCharateristics>
         <ns1:characteristicType listID="token" listAgencyID="102" listAgencyName="String"
                  listName="String" listVersionID="token" name="String" languageID="en-us"
                  listURI="http://www.altova.com"
                  listSchemeURI="http://www.altova.com">token</ns1:characteristicType>
         <ns1:characteristicValue unitCode="E74">0.0</ns1:characteristicValue>
         <ns1:milkingType languageID="en-us">String</ns1:milkingType>
      </ns1:milkCharateristics>
      <ns1:milkingStartingDateTime>2001-12-17T09:30:47Z</ns1:milkingStartingDateTime>
      <ns1:quaterMilking>
         <ns1:quarterIdentifier listID="token" listAgencyID="268" listAgencyName="String"
            listName="String" listVersionID="token" name="String" languageID="en-us"
            listURI="http://www.altova.com"
            listSchemeURI="http://www.altova.com">token</ns1:quarterIdentifier>
         <ns1:quarterMilkingDuration unitCode="NT">0.0</ns1:quarterMilkingDuration>
         <ns1:quarterMilkingWeight unitCode="KPH">0.0</ns1:quarterMilkingWeight>
         <ns1:milkCharacteristics>
            <ns1:characteristicType listID="token" listAgencyID="186" listAgencyName="String"
               listName="String" listVersionID="token" name="String" languageID="en-us"
               listURI="http://www.altova.com"
               listSchemeURI="http://www.altova.com">token</ns1:characteristicType>
            <ns1:characteristicValue unitCode="D78">0.0</ns1:characteristicValue>
            <ns1:milkingType languageID="en-us">String</ns1:milkingType>
         </ns1:milkCharacteristics>
```

```
        <ns1:milkSample>
          <ns1:sampleIdentifierType schemeID="token" schemeName="String"
            schemeAgencyID="85" schemeAgencyName="String" schemeVersionID="token"
            schemeDataURI="http://www.altova.com"
            schemeURI="http://www.altova.com">token</ns1:sampleIdentifierType>
          <ns1:rackNumber>0.0</ns1:rackNumber>
          <ns1:bottlePosition>0.0</ns1:bottlePosition>
          <ns1:bottleIdentifier schemeID="token" schemeName="String" schemeAgencyID="5"
              schemeAgencyName="String" schemeVersionID="token"
              schemeDataURI="http://www.altova.com"
              schemeURI="http://www.altova.com">token</ns1:bottleIdentifier>
          <ns1:validSampleFillingIndicator languageID="en-us">
          String</ns1:validSampleFillingIndicator>
        </ns1:milkSample>
      </ns1:quaterMilking>
      <ns1:animal_id schemeID="token" schemeName="String" schemeAgencyID="147"
            schemeAgencyName="String" schemeVersionID="token"
            schemeDataURI="http://www.altova.com"
            schemeURI="http://www.altova.com">token</ns1:animal_id>
      <ns1:alternative_Animel_Id>
        <ns1:Id schemeID="token" schemeName="String" schemeAgencyID="149"
          schemeAgencyName="String" schemeVersionID="token"
          schemeDataURI="http://www.altova.com"
          schemeURI="http://www.altova.com">token</ns1:Id>
      </ns1:alternative_Animel_Id>
    </ns1:SampleData>
</ns1:provideSampleDataRequest>
```