

Minutes StanForD2010

2012-11-14--15

Place:	Skogforsk, Uppsala 10:00-13:00 CET
Participants:	Mikael Eliasson, LogMax Teemu Paldanius, Ponsse Urban Folkesson, Ponsse Tapio Räsänen, Metsäteho Timo Melkas, Metsäteho Joonas Karnametsä, John Deere Antti Korkiakoski, John Deere Erik Kindlund, John Deere Nazmul Bhuiyan, Skogforsk Jonas Thorsson, Dasa Lars Henriksson, SDC Per Knutsson, Logica (day 1) Vuorenpää Tuomo, Metsähallitus Björn Urbanke, KWF Jean Plamondon FP Innovations (phone) Halvard Olderlund, Komatsu Forest (phone) Fredrik Tjernberg, Komatsu Forest (phone) John Arlinger, Secretary Johan J. Möller, Chairman

§ 1 Opening of meeting

Johan J Möller opened the meeting.

§ 2 Agenda

The agenda was accepted by the meeting.

§ 3 Road map

John presented the suggested roadmap which is a result of three telephone meetings during August and September 2012 (appendix 1).

It was decided to add status type Pre-release to the existing Draft and Release with the following descriptions:

- **Draft** can be used for testing and demonstration at any time by any one.
- **Release** is the final schema version never to be modified, a pre-release becomes a release after a testing period of normally 2-3 months.
- **Pre-release** is only created or modified after a meeting and based on decisions taken at a meeting. Once something has been added to a pre-release we have to take a formal independent decision at a subsequent meeting in order to exclude or modify it. A pre-release becomes a release after a testing period of 2-3 months.

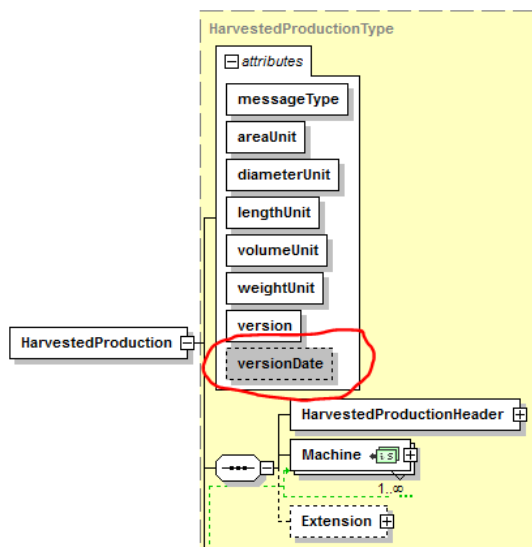
It was suggested to also add some date stamp to the schemas in order to improve management of primarily the draft and pre-release schemas. The main reason for this is that both draft and pre-release versions might be modified more than once.

It was therefore decided to add a new attribute versionDate in two different locations.

The first versionDate was added to all schema headers:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="urn:skogforsk:stanford2010"
xmlns:doc="urn:skogforsk:stanford2010:doc"
xmlns:sfd="urn:skogforsk:stanford2010"
targetNamespace="urn:skogforsk:stanford2010"
elementFormDefault="qualified"
attributeFormDefault="unqualified"
version="release_2.1"
sfd:versionDate="2012-09-06">
```

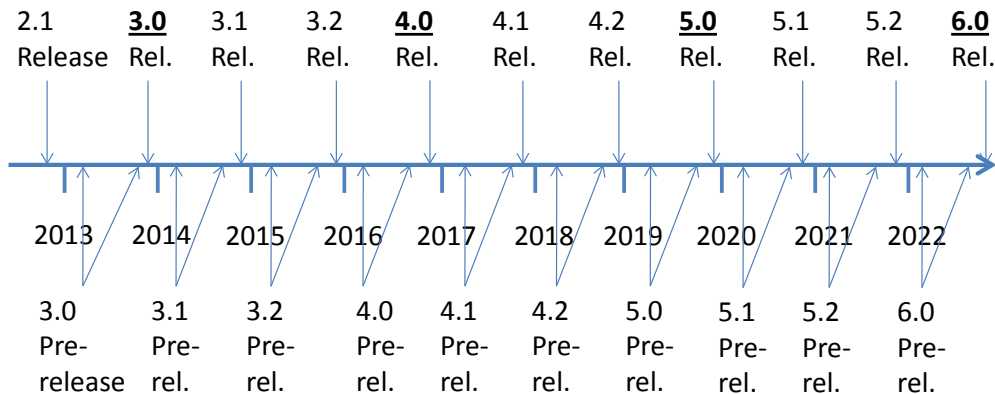
Secondly versionDate was added to attribute list of each message in version 2.1 (will be mandatory in version 3.0):



It was decided that we shall try to follow the following basic rules for updating StanForD2010:

- Never more than one new "release" (minor or major) per year (January or February).
- Maximum number of major releases is once per three years.
- Two meetings per year, in March and October
- All decisions at meetings are included in "pre-release". No other changes allowed in "pre-release".
- A "pre-release" will become a "release" after a two/three month testing period.

The following figure illustrates the plan for new releases and pre-releases:



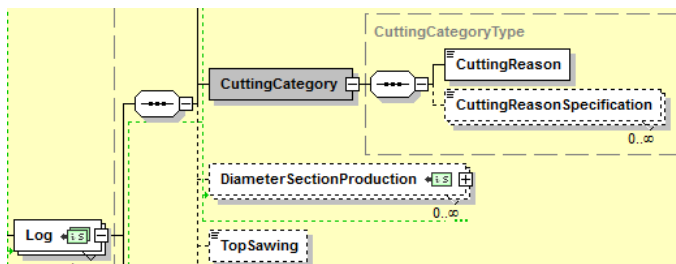
Observe that all decisions at this meeting only affect version 2.1. The first 3.0 Pre-release will be published after the spring meeting 2013 and the final Release version after the autumn meeting 2013.

The question of possible version 2.2 was discussed but no participants expressed any interest in a version 2.2. The secretary expressed that he did not really like the idea of having a version 2.2 parallel with a version 3.0 due to the increasing administrative workload and the risk for errors.

§ 4 Clarifications

USE AND DEFINITION OF CUTTINGREASONSPECIFICATION (ENUMERATIONS: OUTSIDE CUTTING WINDOW, QUALITY CHANGE, UNCLASSIFIED LOG)

It was decided at the last meeting (2012-04-18—19) to update annotation of CuttingCategory and also that Skogforsk should analyze exactly how element CuttingReasonSpecification is to be used (enumerations: outside cutting window, quality change and unclassified log).



The following description is included in var300_t1 in the old standard:

If the log is bucked outside the normal "cutting window" (Swedish "kapfönster") 50 should be added to the code. If the quality change after the bucking 100 should be added. If the log is unclassified 200 may be added to the code.

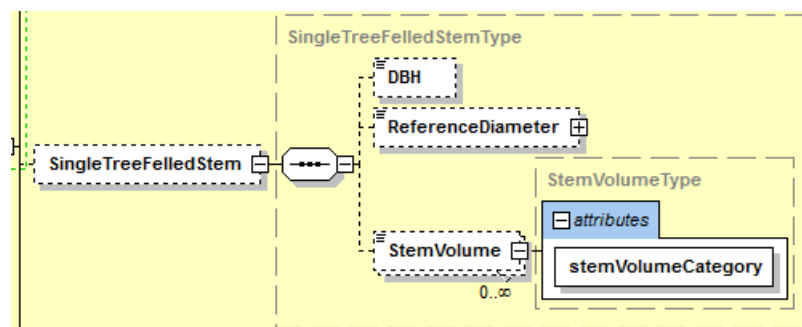
It was decided that the enumerations are defined in the following way (added to annotation):

- Outside cutting window: used if present cut is outside of allowed cutting window for present log.
- Quality change: used if quality is changing after cutting of log.
- Unclassified log: used if present cut means that the present log is unclassified.

It was also discussed at the meeting whether CuttingReasonSpecification is really needed since cutting window, unclassified logs and quality changes are known through other elements. This element may very well be excluded from the 3.0 version.

STEMVOLUME FOR FELLED STEMS. INCLUDE TOTAL VOLUME OR MERCHANTABLE VOLUME?

It is not strictly defined if StemVolume (multi and single tree felling) includes merchantable volume or total volume including top of the stem.



If we look at the old standard and var266_t3 it is clearly stated that stem volume is to include only the merchantable volume (up to a minimum diameter). However since felled stems quite often are very small (outside size of merchantable logs but still useful for energy purposes) it seems to be simpler to include the total stem volume (including top volume)

It was thus suggested that the description of element StemVolume is updated to:

Total volume of stem including also top of stem.

However, it was decided by the meeting to keep the open definition as it is. The possibility of adding an attribute minimumStemDiameter was discussed, this would define what part of the stem that was included in StemVolume.

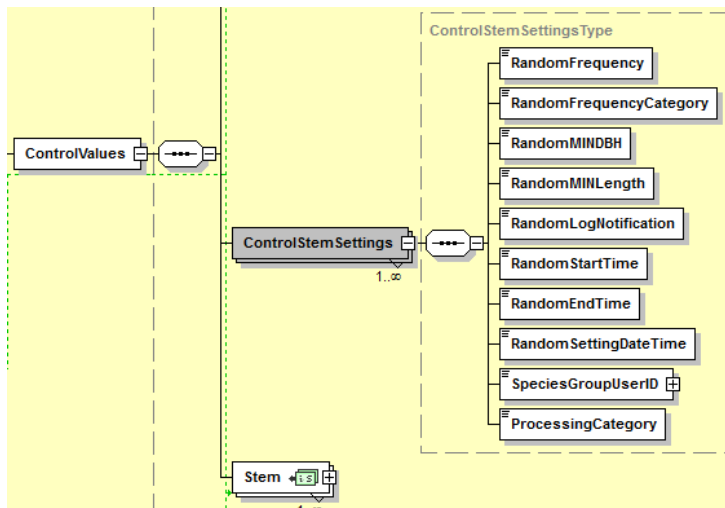
It was decided that the description of attribute stemVolumeCategory is updated to:

Enumeration list: ~~m3 (price) = Volume according to VolumeLengthType, VolumeDiamType and VolumeDiamAdjust (Old log code 400), m3sob = Volume m3sob. (Old log code 401, v256), m3sub = Volume m3sub (Old log code 402),~~ m3sobEstimated = Estimated volume m3sob, m3subEstimated = Estimated volume m3sub.

It was also decided that the annotation is changed from “*positive value*” to “*non-negative decimal*” since this is actually the true type according to the schema.

CONTROLSTEMSETTINGS IN HQC

The proposal is based on some comments from Jonas Thorsson 2012-10-23. The settings for control stems were updated in 2009-11-11 in order to make it possible to register the five last settings per species. This is the reason why element ControlStemSettings is repeatable (figure below).



The following table includes the present annotations for the control stem settings:

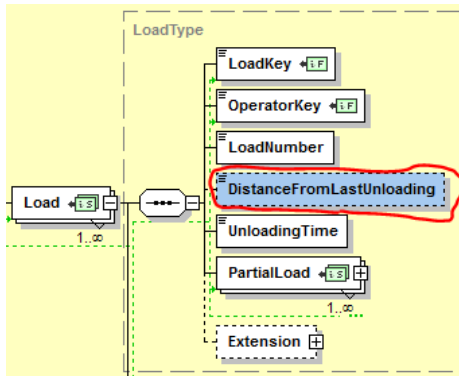
Element	Annotation
ControlStemSettings	Settings for random selection of control stems
RandomFrequency	Sampling interval for random selection of control and calibration see also RandomFrequencyCategory
RandomFrequency-Category	Enumeration list, allowed values are: "Number of stems", "Time" ,"Volume"and "Not in use"
RandomMINDBH	Minimum DBH for randomly selected tree</doc:Description>
RandomMINLength	Minimum log length for randomly selected stems. At least one log within a stem must be at least this long if stem is to be selected as a randomly selected control stem. Operator is never notified that it is a control stem until this criterion is met. Criterion in RandomLogNotification also has to be met before operator is notified. This means that the operator may be notified at a log no higher than set in RandomLogNotification if RandomMinLength is larger than 0. Observe the similarities with RandomMinDbh.
RandomLog-Notification	Log number when operator is notified that the stem has been randomly selected for control purposes. Is to be done when position for cutting of log has been decided but prior to the actual cutting of the log.
RandomStartTime	Earliest time when stem is to be randomly selected, used in order to avoid random selection during the dark hours. This variable must be registered if a timer exist for turning off random selection.
RandomEndTime	Latest time when stem is to be randomly selected, used in order to avoid random selection during the dark hours. This variable must be registered if a timer exist for turning off random selection.
RandomSetting-DateTime	Date and time when the random control settings were last modified in any way. Making it possible to track changes in the settings of random selection of control stems

It was decided that the annotation of ControlStemSettings should be changed to:

Settings for random selection of control stems. Recommended that the last five settings per species group are included in an hqc-file.

DISTANCEFROMLASTUNLOADING IN FPR

The secretary has received questions about definition of element DistanceFromLastUnloading:



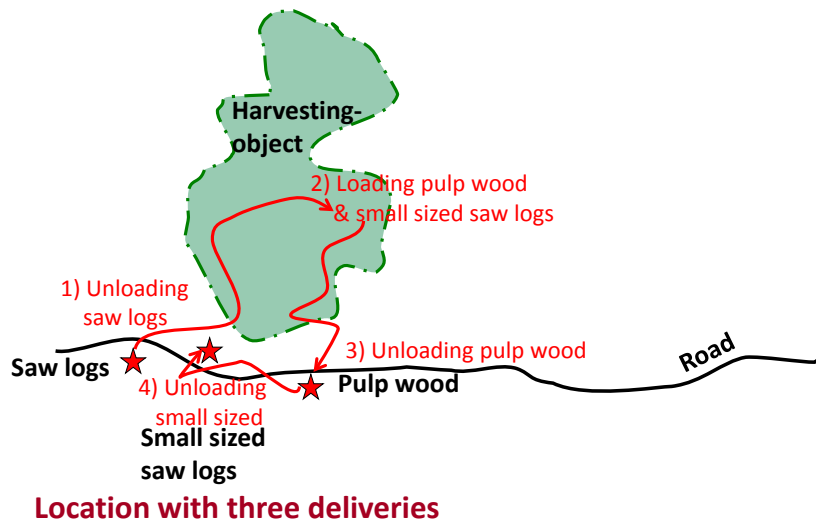
It was decided to update the annotation:

<doc:VersionCreated>1.0</doc:VersionCreated>

<doc:Unit>m</doc:Unit>

<doc:Description>Driven distance from last time of unloading (when load space of forwarder was emptied). Registered when unloading present load. Distance from start of harvesting object is registered for first load. Measured by odometer or gps.</doc:Description>

The following figure illustrates the fact that the total driven distance between point 1 and 4 (red arrows) are to be included in DistanceFromLastUnloading when unloading the pulpwood and small sized saw logs.



The figure above will be included in the document “StanForD2010 introduction”. Observe that this example includes co-loading of two different products.

HARVESTTIME AND STEMCOORDINATE IN HPR. REGISTERED WHEN FELLING OR WHEN PROCESSING IS COMPLETED?

It was suggested that it is clearly pointed out the both HarvestDate and StemCoordinate should be registered at the time of felling of a tree. The coordinates should e.g. not be the position of the machine at the time of cutting the last log of a tree.

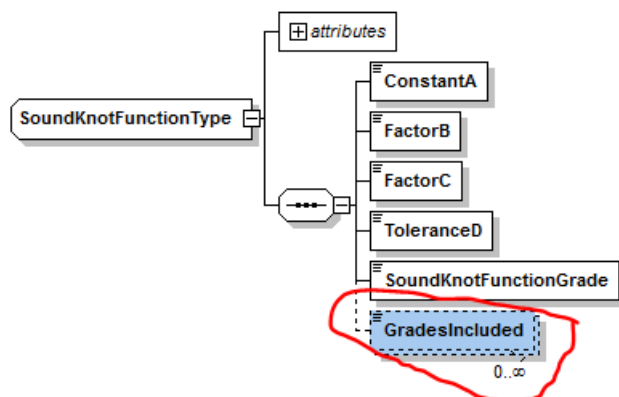
The suggested new annotation was “Stem coordinates, describes the location of the standing tree. To be registered at the time of felling of a stem.” The meeting decided not to update the annotation since the enumeration “Crane tip position when felling the tree” was deemed to cover the needs.

§ 5 New elements and attribute

SOUNDKNOTFUNCTION

This issue was discussed at previous meeting (2012-05-18) where new element GradeExcluded was added.

A new suggestion based on discussion with Anders Nylund, Komatsu (2012-06-27). It was decided to include an element “GradeIncluded” instead of “GradeExcluded”. This means less problems if an operator uses special grades, for example grade 9 for long poles etc, since exactly what special qualities will be used are not always known when creating an spi-file.



The new optional element will have the following description:

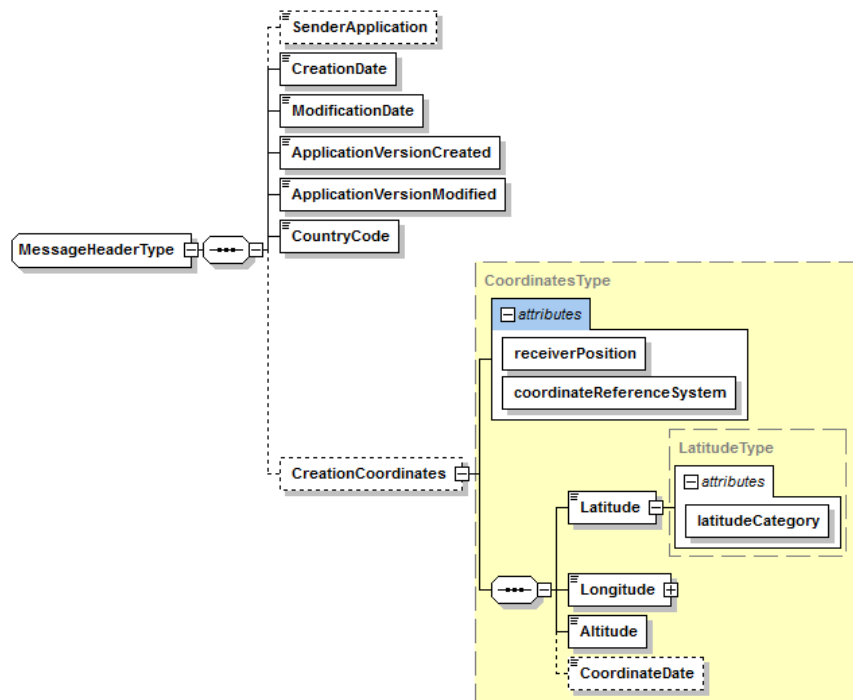
“Grades that can be replaced by the automatically calculated SoundKnotFunction grade. Means that if a section of stem does not have a grade included in this element the SoundKnotFunction will NOT affect the bucking optimization. All grades with a number below or equal to StartGrade and SoundKnotFunctionGrade should always be replaced if element GradeIncluded is missing”.

Coordinates in message header

Proposal from Joonas Kaarnametsä, JohnDeere (2012-09-10).

Joonas would like to see the coordinates in the file header, with the location information stored at the time of file creation. That way it will be easy to get some location information from all machines in all files without the need of browsing through possible stem (HRV) or location/load (FRW) elements in specific types of messages.

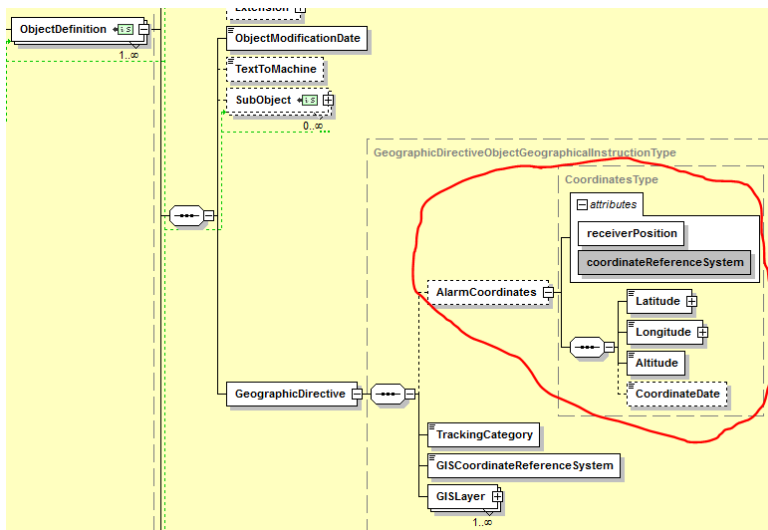
It was decided that CoordinatesType element with the name CreationDate is added as optional in header element of each file (illustrated below) with annotation: “Coordinates of machine when creating file, ie position at the time register in CreationDate.



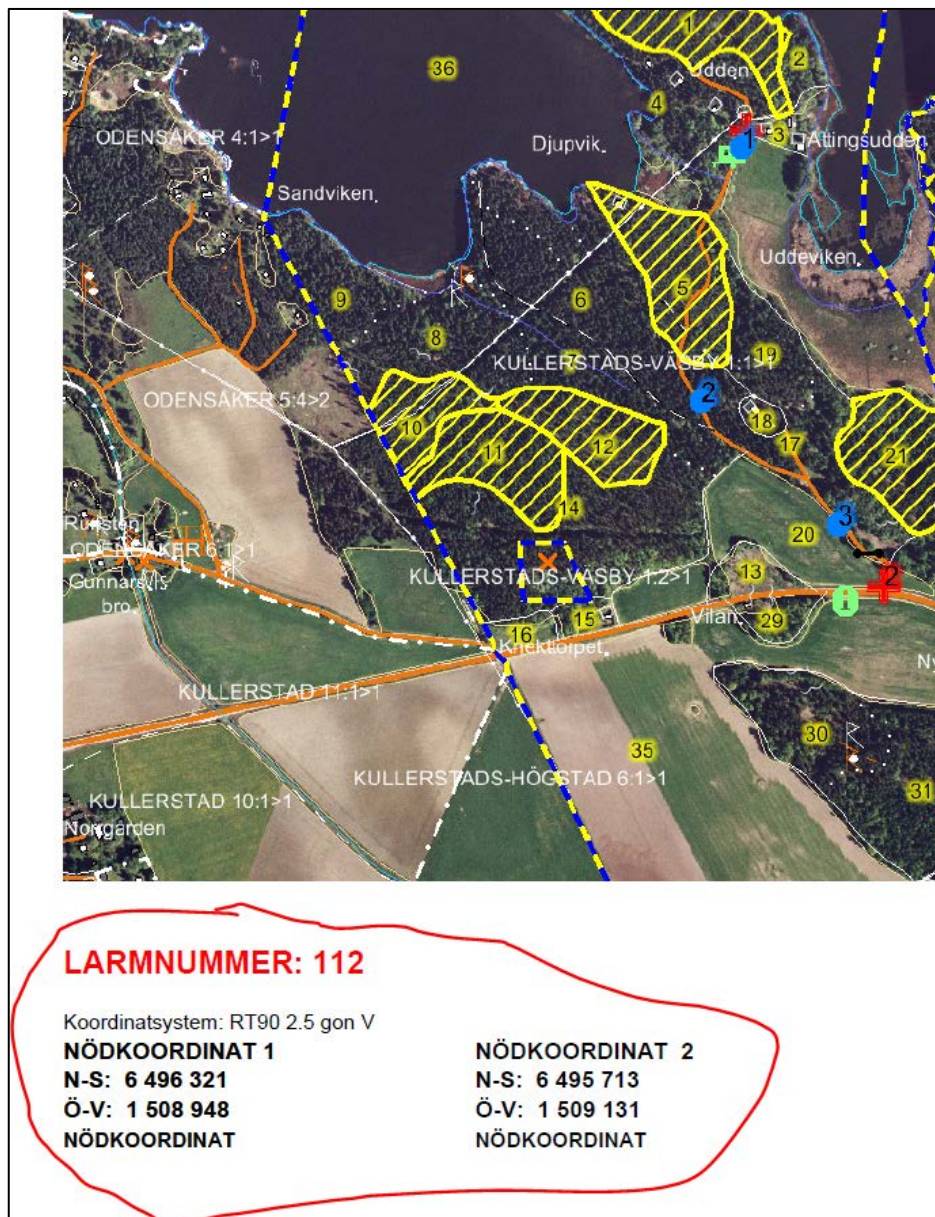
The new element CreationCoordinates will make it possible to answer not only when a file was created but also where it was created.

Alarm coordinates in ogi

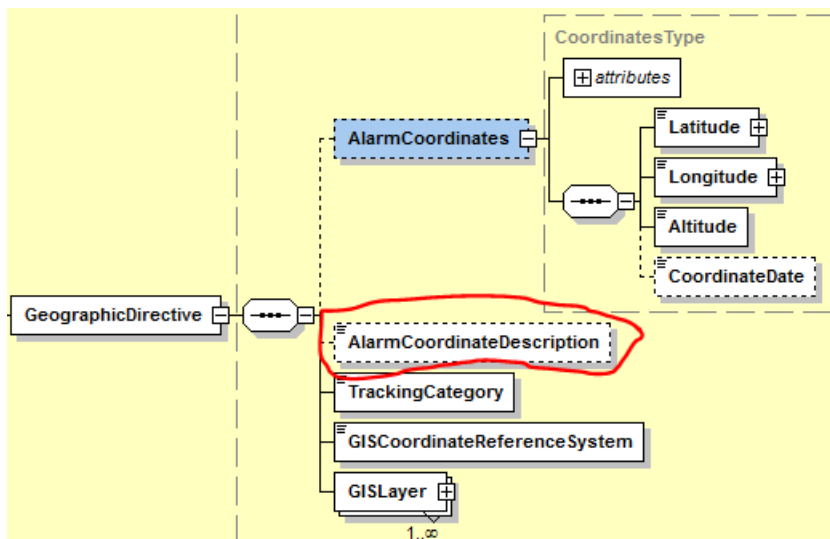
Secretary has received some comments (LogMax, August 2012) regarding Alarm coordinates registered according to figure below.



It is common in Sweden that the alarm coordinates are printed in a pdf-instruction using some local reference system like for example SweRef99 as illustrated by the following figure from a pdf document.



A possible solution could be to include alarm coordinates as free text where any reference system might be used in order to communicate this information to the operator if he e.g. wants to tell the position over the phone.

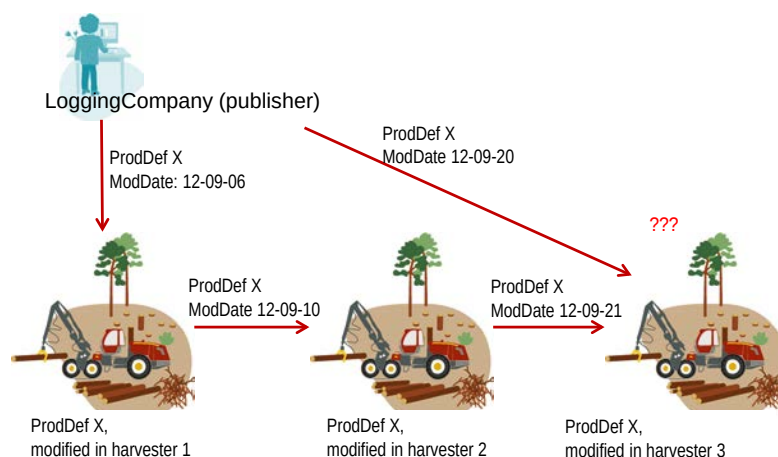


The meeting decided to postpone this proposal. If implemented the element **AlarmCoordinateDescription** should be repeatable.

New attribute modificationAuthor

Suggestion based on a discussion with Fredrik Tjernberg and Niklas Ericsson at Komatsu (2012-09-18). The objective of this proposal is to improve the user interface in the machines when receiving new instructions (e.g. products, objects, species groups etc).

It was discussed how to use **ModificationDate** in cases where a **ProductDefinition** is sent to a harvester and then modified in that harvester or another harvester several times. This case could be described in the following figure:



The only thing stated clearly is that if both **ProductUserID** and **ModificationDate** are the same there is no need at all for updating the “product database” in the harvester. The operator must always have the final decision when one product in the “database” is to be

replaced even if the product in the machine is younger than the new one sent to the machine.

By introducing an optional free text attribute, where it is stated who did the last modification, it is possible to inform the operator not only about when the product was last modified but also the author of these changes.

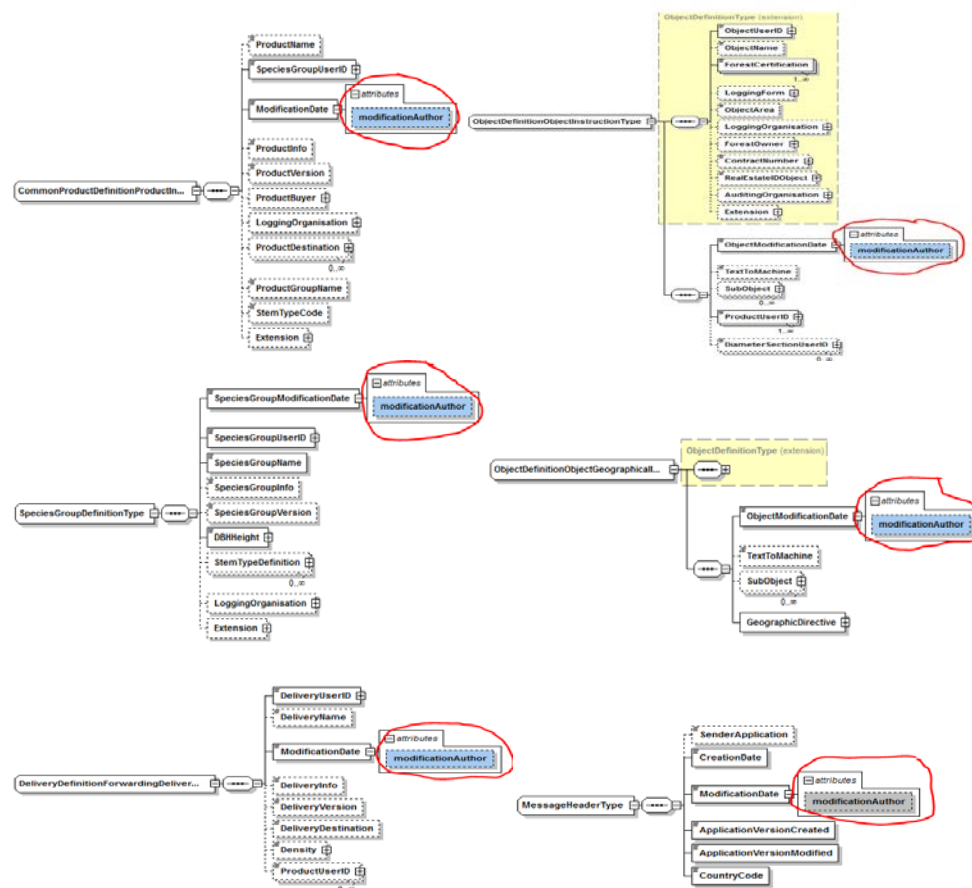
The operator could for example get the following question when importing a new product definition:

"Do you want to replace present product "Pine small sized timber" with product "Pine small sized timber"?

The present version was modified by "Holmen skog" at 2012-09-20 12:15:02.

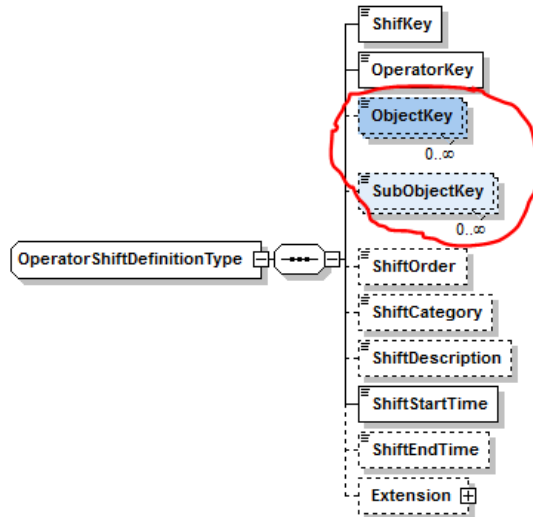
The imported version was modified by "MaxiXplorer 941.1" at 2012-09-21 09:27:39.

Below are a number of figures illustrating where this new decided attribute (optional free text) should be added. Observe that it should be added to all ModificationDate elements. The annotation is *Author responsible for the last modification*.



Object- and SubObjectKey in ShiftDefinition

It was suggested by Niclas Eriksson, Komatsu, 2012-08-30, to update the ShiftDefinitionType (mom-file) in order to register more than one object on a shift. It was suggested to solve this by making ObjectKey and SubObjectKey repeatable:



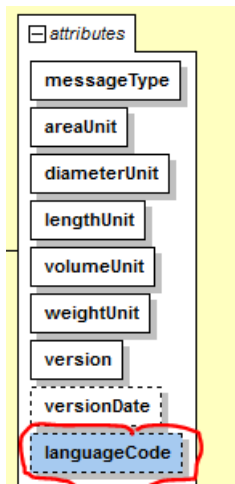
The meeting decided not to update the standard since the ObjectKey and SubObjectKey can be found under OperatorWorkTime where we also have a reference to one shift through a ShiftKey.

Language attribute in message header

Proposal from Tapio Räsänen (2012-09-27).

In the Finnish StanForD meeting (September 2012) a proposal to add a new attribute to “User defined data” for language information was discussed. The idea was that the attribute would tell the language of the message content: Finnish, Swedish, English and Portuguese etc. This would help in the control of the user interface. The attribute could be added to the root level together with messageType, areaUnit etc. and the enumerations should be according to ISO standard.

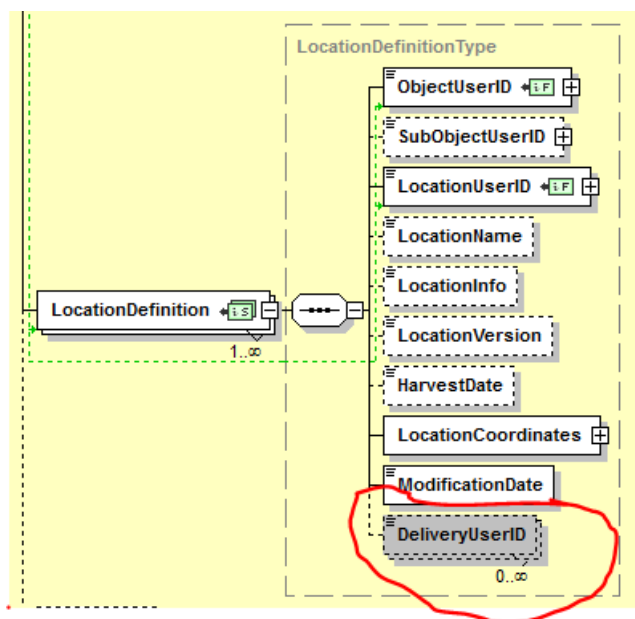
It was suggested to add optional attribute languageCode with the following definition to all schemas: *The ISO language code that describes the language that is used in message content. Eg. fin (Finnish), swe (Swedish), fra (French), deu (German), Unit is ISO 639-3 code*



The meeting decided to postpone this issue since there was no clear description of how to use it.

UPDATING LOCATIONDEFINITION

Proposal Henrik Persson LogMax (2012-10-19). Make it possible for forest company to define what Deliveries to use on a certain location (landing) by adding DeliveryUserId to LocationDefinition.

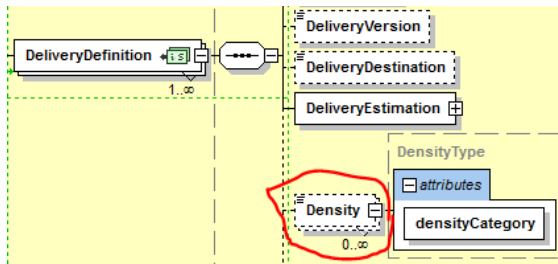


Proposal withdrawn by LogMax but some participants found it to be of interest. The issue will probably be discussed again at the next meeting since several participants found the idea to be relevant.

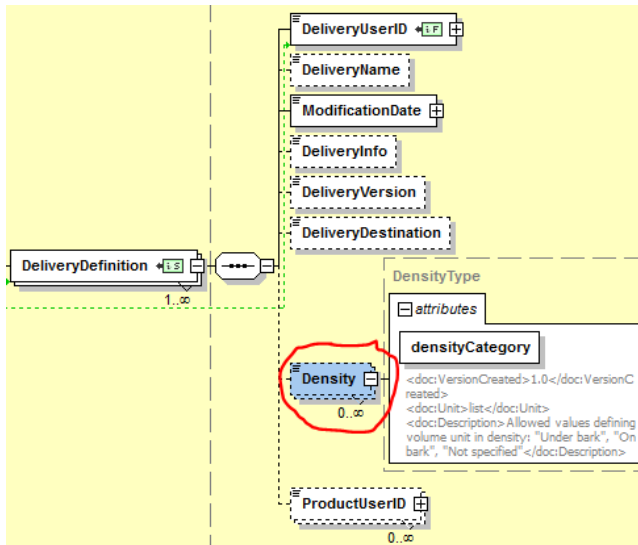
DENSITY IN FORWARDING DELIVERY INSTRUCTION

Proposal from Teemu Paldanius, Ponsse 2012-10-26.

Element density was made repeatable in fpr message at the last meeting (see below) in order to include several different types of green densities (under vs on bark).

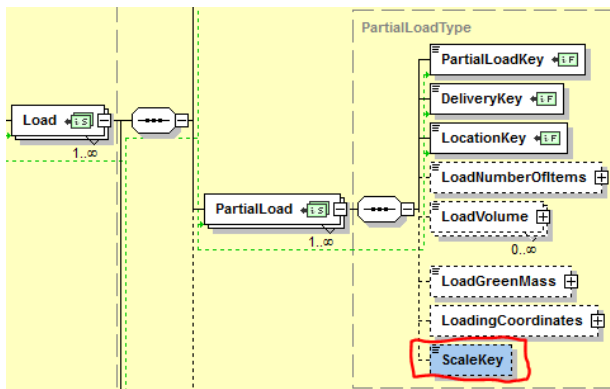


It was therefore decided to make the Density element repeatable also in the fdi message according to the following figure:



SCALEKEY IN LOAD STRUCTURE OF FPR-MESSAGE

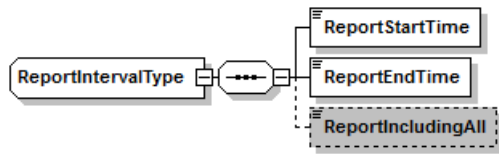
Original proposal from Teemu Paldanius, Ponsse (2012-08-14). It was decided that ScaleKey is added as an optional element to LoadType. That way it is possible to connect each load to a certain scale device (for example if scale is changed).



FILTERING OF MOM MESSAGES (REPORTINCLUDINGALL)

A new optional boolean element "ReportIncludingAll" was added 2012-04-18 (observe that it will be optional in a new version 2.1) with the definition "All run (work) times, down times and meal breaks that occurred between ReportStartTime and ReportEndTime are

included, with no exceptions. This means that any times not registered are to be considered to be unutilized”.

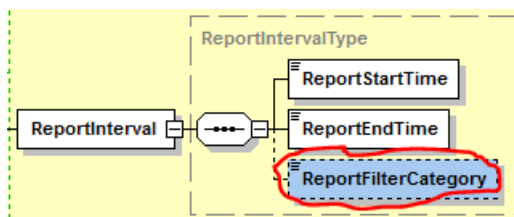


This means that times not being registered, in case of a time oriented report, are all to be considered to be unutilized.

New suggestion:

It was decided that the element name is changed to ReportFilterCategory with the following enumerations: No filter, Object filter and Other filter.

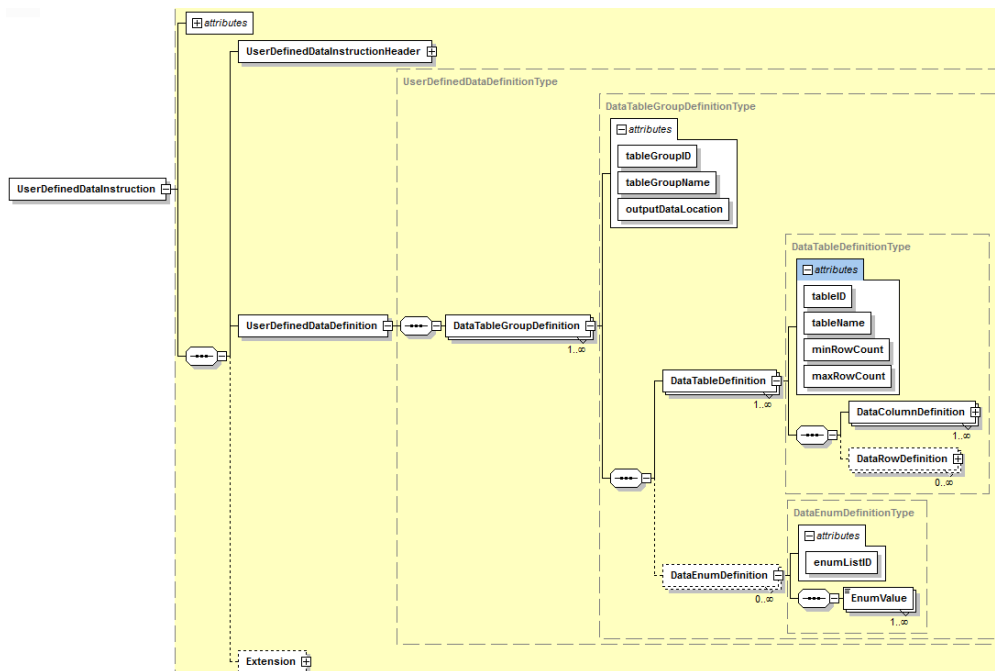
Observe that “No filter” means that “All run (work) times, down times and meal breaks that occurred between ReportStartTime and ReportEndTime are included, with no exceptions. This means that any times not registered are to be considered to be unutilized.” The value “Object filter” means that only times directly connected to a specific object are included.



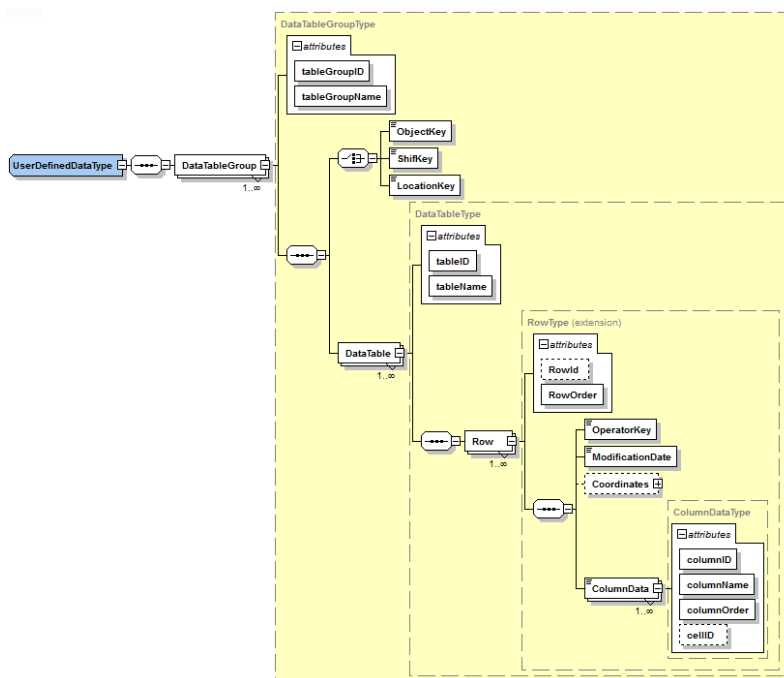
§ 6 User defined data

A complete description of “user defined data” can be found in appendix 2.

The meeting decided to include a new message User defined data instruction (udi) in version 2.1 with the main structure illustrated below:



The meeting also decided to include the structure `UserDefinedData` in `hpr`, `thp`, `fpr` and `mom` according to the following structure:



§ 7 Finnish proposals

Three major Finnish forest companies (StoraEnso, Metsähallitus, Metsä) have decided to jointly develop a system called “Common planning system (KSP)”. StanFord2010 has been analysed based on the preliminary needs of such a system and a number of questions and comments have therefore been raised. The aim is to base the planning system on version 3.0. Therefore the main part of required changes to the standard should be found

and defined already during the coming winter. Present understanding of the KSP work group is that the most critical questions can be specified by the next StanForD meeting, but probably there will be minor needs for changes still after that when the building work of KSP system and forest companies' ERP systems really starts. Questions related to control of bucking and use of Apter software will be the first ones to decide on. These different issues are all included in appendix 3. The conclusions from the meeting discussions are that some issues can be solved by updating the basic standard while other issues only need clarifications of the present standard and the ideas behind the standard. No decisions were made during the meeting. Skogforsk will cooperate closely with Metsäteho and the Finnish forest companies in order to write concrete proposals and clarifications for the next meeting.

§ 8 Version 3.0 issues

The following issues all relate to a coming version 3.0. No decisions were taken at the meeting since 3.0 (pre-release) will not be implemented until spring 2013.

MANDATORY REPORTFILTERCATEGORY IN MOM

The previously decided element ReportFilterCategory should be made mandatory in 3.0.

STRUCTURE OF CLASSIFIEDPRODUCTDEFINITION

The implementation of classified product definition structure in StanForD 2010 schemas is not optimal if we want to support reusability and avoid redundancy in data structures. The following table gathers all types that define classified product specific information and in which messages they are used. The first three types are defined in StanForD2010CommonDefinitions file. All the other types are message specific. We can almost immediately see that same information is defined in many places. For example the common definition type is nearly identical and it can be found in three different files: one in StanForD2010CommonDefinitions file and the others in fdi and pin messages.

Type/Message	fpr	fdi	hpr	hqc	pin	thp
CommonProductDefinitionType	x		x	X		x
CommonProductDefinitionMachineType	x		x	X		x
ClassifiedProductDefinitionMachineType				X		x
ClassifiedProductDefinitionMachineHarvestedProductionType			x			
CommonProductDefinitionProductInstructionType					x	
ClassifiedProductDefinitionProductInstructionType					x	
CommonProductDefinitionForwardingDeliveryInstructionType		x				
ClassifiedProductDefinitionForwardingDeliveryInstructionType		x				

This means that if we want to, for example to update CommonProductDefinitionType in StanForD2010CommonDefinitions file we have to update also types in both pin and fdi messages. This kind of data redundancy is not desirable. This is not only a schema management problem because applications that are processing messages also have to be updated in many different places. Therefore we suggest that we reorganise product structures in schemas so that redundant parts can be removed as much as possible.

Important thing is to analyse what are actually the differences between classified product definitions in separate messages? There is naturally common parts that are equal in several messages. The distinguishing parts (elements and structures behind them) have been presented in the following table:

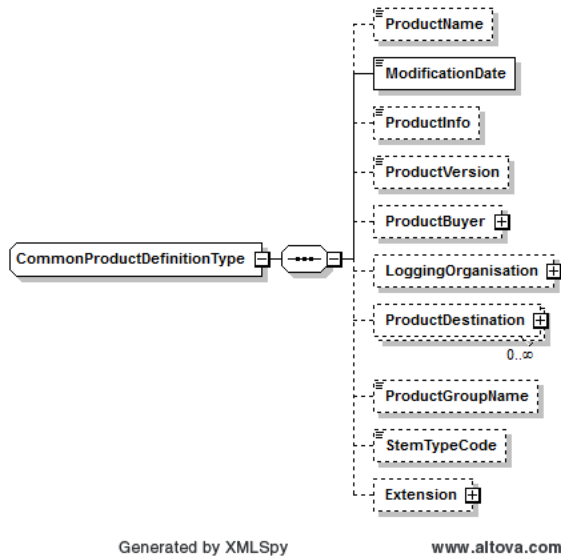
Element/Message	fpr	Fdi	hpr	hqc	pin	Thp
ProductUserID	x		x	x		X
SpeciesGroupUserID		X			x	
SpeciesGroupKey	x		x	x		X
DiameterDefinition			x	x	x	X
LengthDefinition			x	x	x	X
PriceDefinition			x	x	x	X
ProductCreationDate		X	x		x	
ProductPresentationOrder		X	x		x	
ManualFreeBuck			x		x	
TopLogFreeBuck			x		x	
ProductCondition			x		x	
CuttingWindowDefinition			x		x	
PermittedGradesDefinition			x		x	
LengthDistributionDefinition			x		x	
LimitationDefinition			x		X	
ProductMatrixes			x		X	
LongLogButtHeight			x		X	

We can see from the table that identification information varies quite a lot between different messages. However, there are two sets of elements that are included in multiple messages. Elements inside dark red rectangle exist in four messages and elements inside light red rectangles exist in two messages. We can use this information to provide reusable structures that are defined only once in Stan-ForD2010CommonDefinitions file.

We therefore suggest that we reorganise product structures in schemas so that most of the redundant parts are removed.

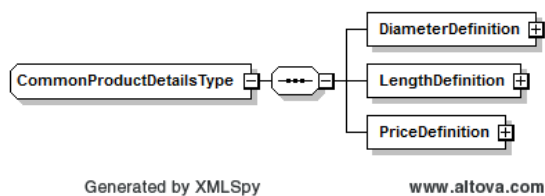
The suggestions are:

- 1) We provide one type definition where we gather all information that is common for all classified products. That kind of type already exists in StanForD2010CommonDefinitions, that is CommonProductDefinitionType.

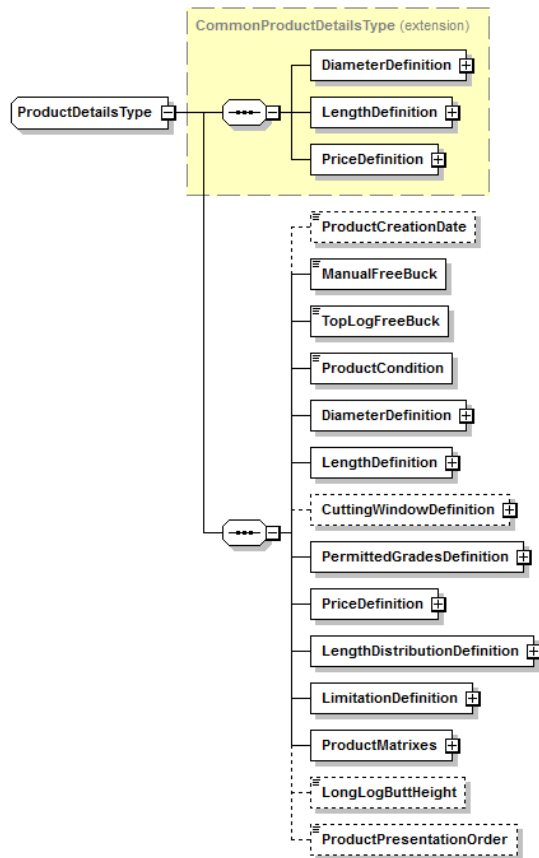


2) Then we provide subtypes for each message that has classified product structure. All those subtypes inherit CommonProductDefinitionType and additionally they define few elements that are varying. Elements that have to be defined in each message are the ones that have no color background in table above. Additionally we add one mandatory element ProductDetails for those subtypes (messages) that contain elements that are colored in table above. Those new element structures are described in 3) and 4). These subtypes are message specific and therefore they are defined in schema files of corresponding messages.

3) We include in StanForD2010CommonDefinitions a totally new type CommonProductDetailsType that has three elements (the ones that are dark red in table):



4) We then include a new subtype ProductDetailsType that inherit CommonProductDetailsType in StanForD2010CommonDefinitions file.

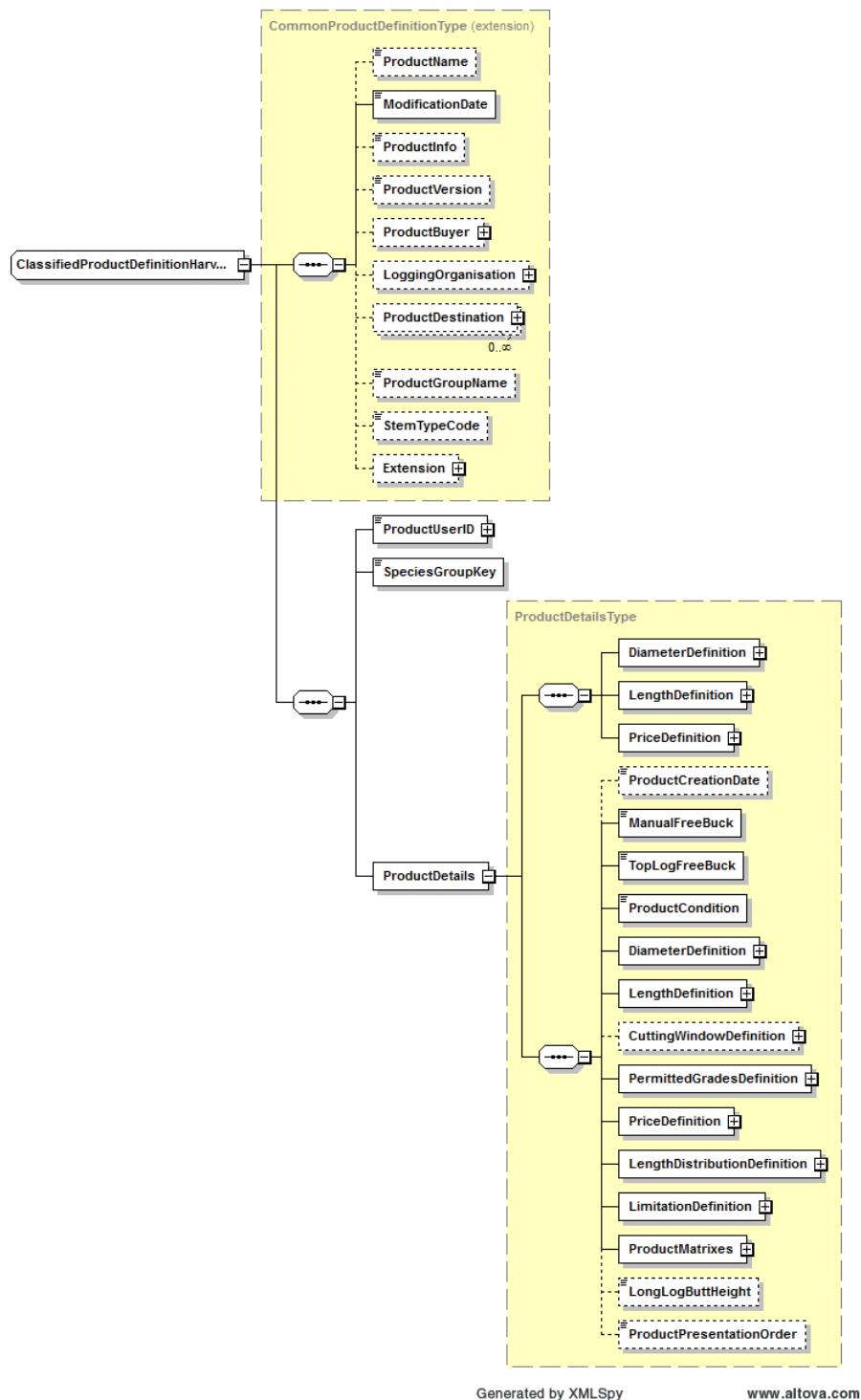


Generated by XMLSpy

www.altova.com

Types defined in 3) and 4) are used in message specific classified product structures as needed. For example the **ClassifiedProductDefintionHarvestedProductionType** in hpr

schema would be:



No objections were raised at the meeting concerning this proposal. The issue of whether to use SpeciesGroupUserID instead of SpeciesGroupKey in ProductDefinition (in hpr-fiels) was raised by Teemu. Jonas Thorsson stated that he thought that SpeciesGroupKey in ProductDef (pin) and in DiameterSectionDefinition (spi) should be excluded in version3.0.

This issue was also discussed at the last meeting without any decisions to change the structures since some manufacturers already has implemented the present solution. The meeting did not decide to do any changes.

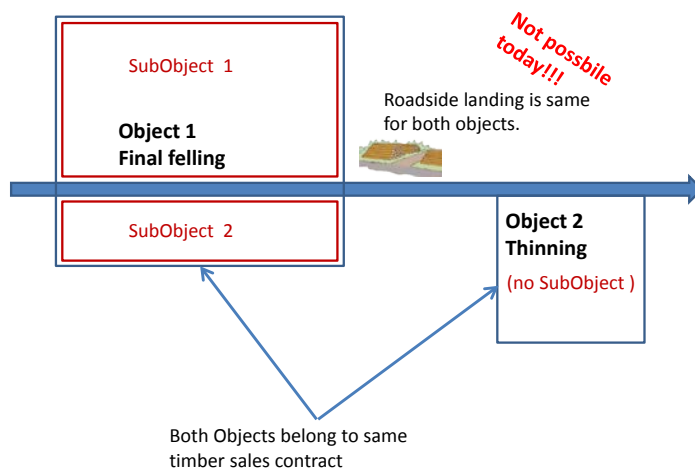
FORWARDING OBJECT INSTRUCTION

A timber sale contract can, in Finland, have several objects. An object is a logging area having the same logging form (final felling, first thinning etc.) and one single harvesting instruction. An object may consist of sub-objects which are normally the forest compartments of the forest plan. Sub-objects are needed to keep harvested volumes of different stands separated for planning purposes.

If objects are close to each other they have only one roadside landing. Piles for different assortments are located at the landing. Timber from all the objects can be forwarded into the same assortment-specific pile. That is why several objects can have one common location.

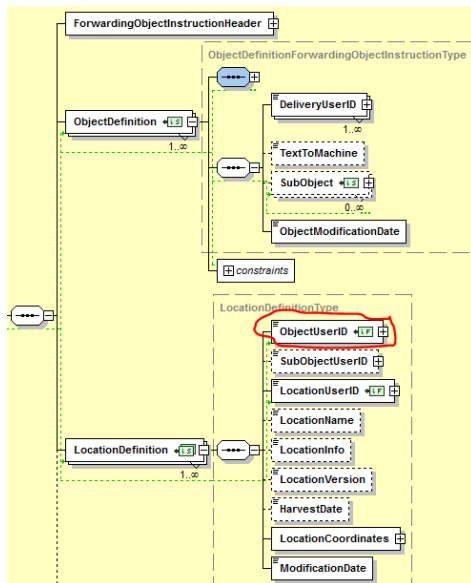
This is a Finnish proposal based on work done in the project "Common planning service for contractors" (companies involved are Stora Enso, Metsä Group and Metsähallitus)). The issue of updating the forwarding object instruction (foi) was first raised in July 2012.

An example illustrating why one single landing may sometimes be used for several different objects is shown in figure below. A landing is presently identical with a LocationDefinition in foi- and fpr-files.



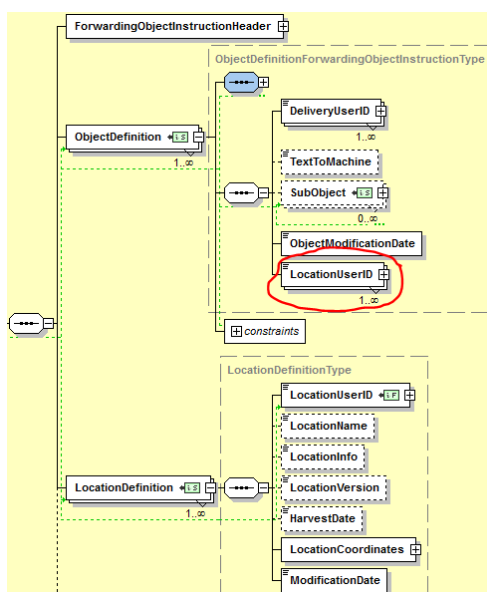
Example illustrating the need for using the same road side landing on several objects.

A problem with the present foi schema is that a specific Location (landing) can only be connected to one single object since ObjectUserID is not repeatable in LocationDefinition (figure below).



Present structure.

A suggested update making it possible to use the same location (landing) for different objects is illustrated in figure below. The difference is that a repeatable LocationUserID has been added to the ObjectDefinition.

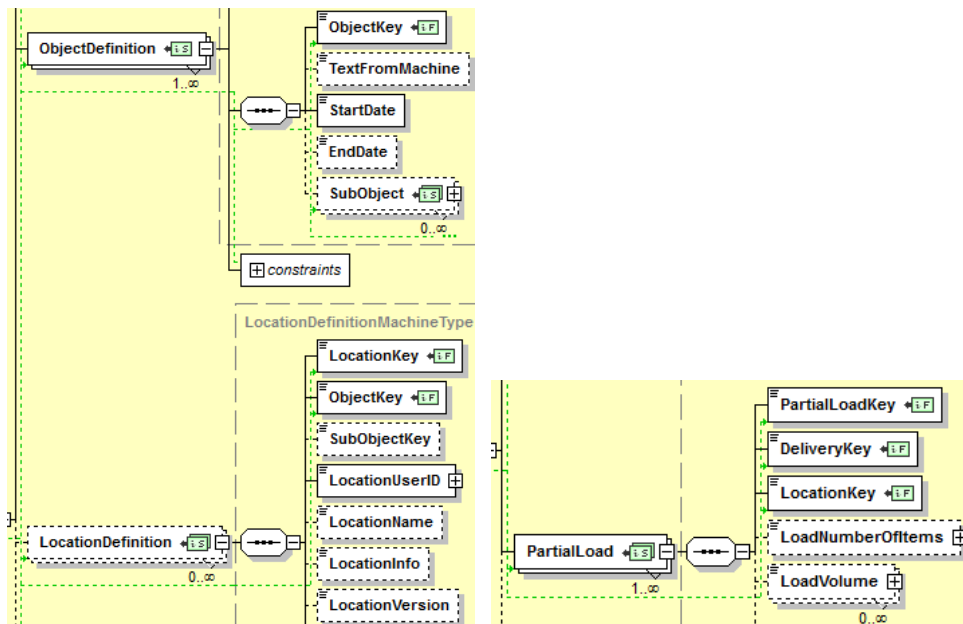


Suggested new structure.

Juha-Antti Sorsa presently does not think that there is any need for changing the fpr. This issue should be further studied and some examples created.

However if we look at the present structures in the fpr-files (figures below) it can be noted that only one single ObjectKey (not repeatable) can exist per Location. If we made that ObjectKey repeatable it we still do not know from which Object a certain PartialLoad originally came from since the PartialLoad only refers to a Location and not to an Object.

It was discussed at the meeting to add ObjectKey and SubObjectKey to Load (not necessarily to PartialLoad) since this may solve the problem described above.



Alternative solution

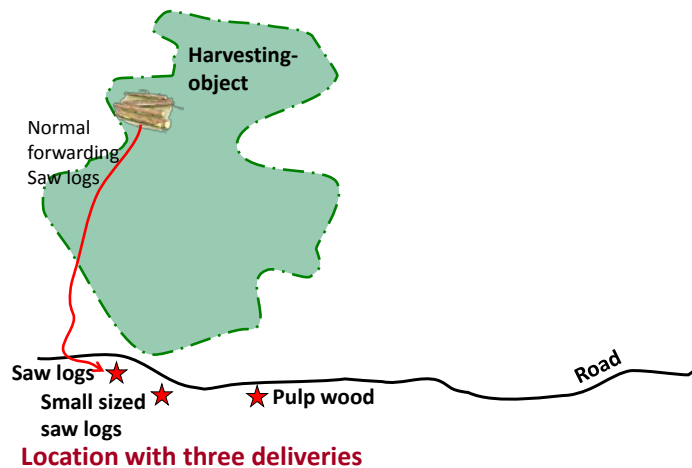
A possible way of handling the need for having the same Locations at more than one Object could be to create two separate Locations with the same identities and coordinates but different LocationUserIDs. Whether this is possible depends on the company systems generating the instructions and receiving the production data.

Object-UserID	Location-UserID	Location-Name	Location-Info	Location-Version	Location-Coordinates
Object_1	Gdf234_1	North	32453	H2012_Sept-20	60,12; 13,24
Object_2	Gdf234_2	North	32453	H2012_Sept-20	60,12; 13,24

PRODUCT MIGRATIONS DURING FORWARDING

The issue of how to handle product migrations (sortimentsvandring in Swedish) in forwarded production message (fpr) was first discussed during a meeting in Helsinki 2012-10-14. The issue was raised by Tuomo Vuorenmaa, Metsähallitus.

Normally forwarding means that a harvested product is transported to a corresponding Delivery at a Location. For example, saw logs are normally forwarded to a Delivery intended for saw logs as illustrated below:



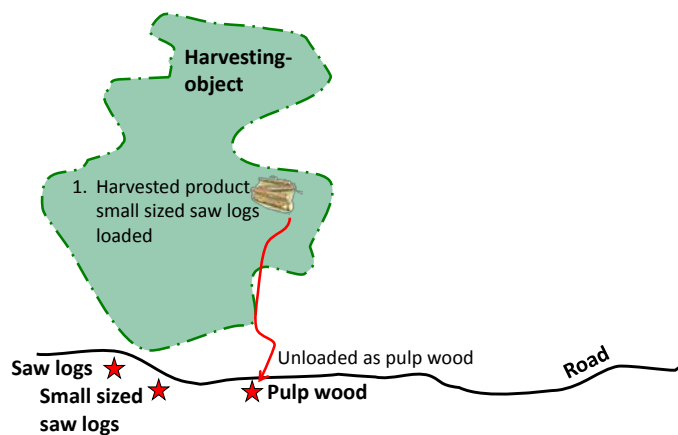
However, in some cases a harvested product may for different reasons be forwarded to a diverging Delivery. This is what “product migration” is all about. It is an important issue when planning the transportation from the forest to the industry.

Product migration during forwarding could be divided up into two different types:

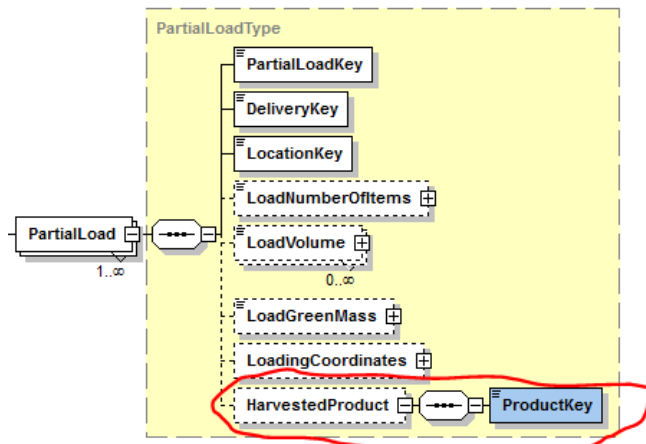
1. Harvested product merged into deviating Delivery
2. Volume at one Delivery/Location moved to different Delivery/Location

Harvested product merged into deviating Delivery

This type of migration occurs when a harvested product is forwarded to a completely different Delivery. For example if saw logs are forwarded to a Delivery including only pulp wood which may occur if the harvested volume of saw logs is too small:

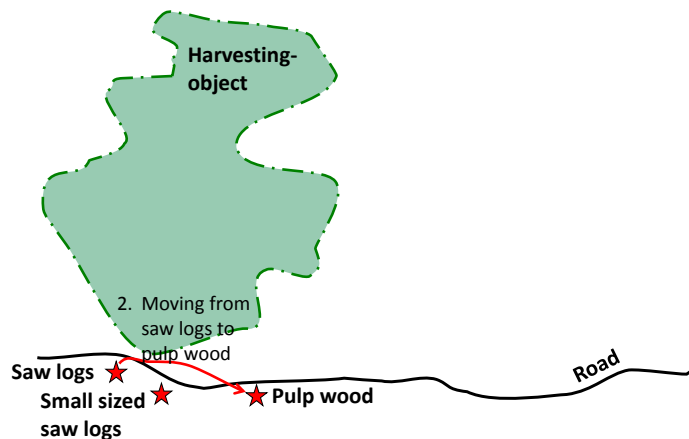


A possible solution for making it possible to register this kind of migration could be to add an optional element HarvestedProduct:

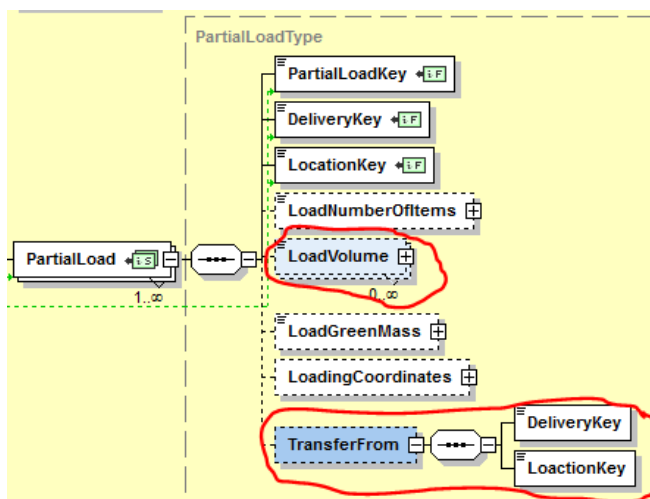


Volume moved between different Delivery/Location

This type of migration occurs when a forwarded volume is moved from one Delivery to another Deliver. For example if saw logs are moved to a Delivery for pulp wood due to the fact that the forwarded volume of saw logs were too small:



A possible solution for making it possible to register this kind of migration could be to add element **TransferFrom** (or perhaps to allow negative **LoadVolume**):



Conclusion

It was pointed out by both Erik and Urban that there is an operator need for making adjustments of the forwarded volumes. For example if there has been a manual error adding 100 m3 instead of 10 m3 of a certain product. The conclusion of the secretary was that the simplest solution to both product migrations as well as operator errors is to allow negative volumes in “LoadVolume”. Maybe also adding an additional attribute telling that some kind of “adjustment volume” was registered by the operator.

§ 9 Postponed issues from spring meeting

The following issues were discussed at the previous meeting (2012-04-18—19) but Skogforsk has not had the time to write any proposal for this meeting:

- Traceability of products in use at an object
- Registering double stem/top.
- Tree codes for high stumps and stump treatment.
Both German and Finnish participants mentioned during the meeting the interest in being able to add “tree codes” for windthrows and beetle infected trees.

§ 10 Updated Swedish reporting recommendations

The proposed new reporting recommendations were presented by the secretary (appendix 4). The only significant comment at the meeting was that the old requirement on having the possibility to re-create partialy reported hpr-files is not necessary anymore. The secretary agreed.

§ 11 KWF project “ForstInVoice”

Björn Urbanke informed about the upcoming KWF project “ForstInVoice”. The project will include the use of “user defined data”.

§ 12 StanForD2010 test

Johan presented some preliminary results and informed about the planning for publishing the results.

§ 13 Development of SilviA for StanForD 2010

Per Knutsson reported that a first test with SilviA7 has been carried out with a user group in Sweden. On the 6th of November they together produced harvester instructions with SilviA 7 and then tested them on John Deere harvester simulator. There is still a beta version and Logica will develop the version a little bit more before it will come out at the market during the spring of 2013.

§ 14 Information material regarding StanForD2010

Johan J. Möller presented the brochures regarding StanForD 2010. Skogforsk will soon print a limited number of these brochures. Contact Skogforsk as soon as possible if you want paper copies or if you have any further comments.

§ 15 Skogforsk project “Calibration of MTH volumes”

Johan J. Möller presented the plan for a Skogforsk project where a method for calibrating MTH-volumes will be developed and tested.

§ 16 Discussion about updating the old standard and length of change period.

This is the first meeting where no request for updating the old StanForD standard has been received by the secretary. However, we will keep the possibility open for at least one more meeting.

§ 17 Other issues

John Arlinger informed about the fact that at least one major Swedish forest company is planning to update the planning system in order to start using the ogi-file (geographical object instruction).

§18 Next meetings

2013-02-05 Phone meeting at 09:00 CET (confirmation of release 2.1).

2013-04-10--11 Regular StanForD meeting in Uppsala .

§ 19 End of meeting

Johan thanked everyone for their participation.

Roadmap StanForD2010

Background

The issue of when to update the standard from one major version to another has not really been discussed in any detail within the StanForD2010 working group. No recommendations concerning the decision process and time tables have been specified.

During the summer of 2012 some updates that required a new major version 3.0 was discussed. It became evident that upgrading of major version is an important issue that has to be clarified since the reactions to this proposal differed significantly between different manufacturers. To a certain extent the positive or negative responses depended on what development stage the different manufacturers were in.

It was decided (2012-08-20) to write this document describing a road map for the foreseeable future regarding how to update StanForD2010. Having clear recommendations and a road map will hopefully mean that all actors can adjust their own development work accordingly and thus avoid potential conflicts regarding when to implement new standard versions.

Schema status

A corner stone when it comes to versioning of StanForD2010 is the fact that we have included the status of a certain version in the schemas. The attribute “version” therefore may have either of the following status values: `version="draft_2.0"` or `version="release_2.0"`. The “release” status means that no changes are ever to be made to this schema version while “draft” versions can be changed and are often used for testing or illustrating new proposals. The “draft” has so far also been used for communicating what was decided at a specific meeting. This means that the draft is used both for communicating decisions as well as for testing purposes making this type of version a bit difficult to interpret.

In order to improve the development of new versions it was suggested by Juha-Antti Sorsa that we should also add one more status level to the standard in the form of a “pre-release”.

The “pre-release” should be a version that manufacturers and developers

- Could start to implement in their systems
- Can be sure that possible changes to that version in forthcoming meetings are not significant.

The new status type would make it evident what was the result of previous meeting vs results of different proposals being discussed in-between meetings.

A “pre-release” is used only in cases where we have a formal decision (at a meeting) to implement something. These decisions are normally not to be changed at next meeting even if the version number has not changed. If we e.g. decide on new structure for ClassifiedProductDefinition at a meeting we should normally not discuss this issue again at the next meeting even if it is the same version at both meetings. This is of course unless someone raises some specific objections to a decision after the meeting.

Schemas published with “pre-release” status must have a testing period (2-3 months) before final release is published. All members must verify that the Pre-release is ok and that the “release” therefore can be published.

It was thus decided that we implement the following status levels in StanForD2010 (2012-11-14):

- **Draft** can be used for testing and demonstration at any time by any one.
- **Release** is the final schema version never to be modified, a pre-release becomes a release after a testing period of normally 2-3 months.
- **Pre-release** is only created or modified after a meeting and based on decisions taken at a meeting. Once something has been added to a pre-release we have to take a formal independent decision at a subsequent meeting in order to exclude or modify it. A pre-release becomes a release after a testing period of 2-3 months.

Below is the updated rule from StanForD_2010_naming_and_design – document.

[Rule 8.2] The xsd:schema version attribute MUST use the following template:
 <xsd:schema ... version="draft" | "pre-release" | "release"<major>.<minor> >
 Where:
 draft | pre-release | release – is used based upon the status.
 <major> - sequential number of the major version.
 <minor> - sequential number of the minor version.

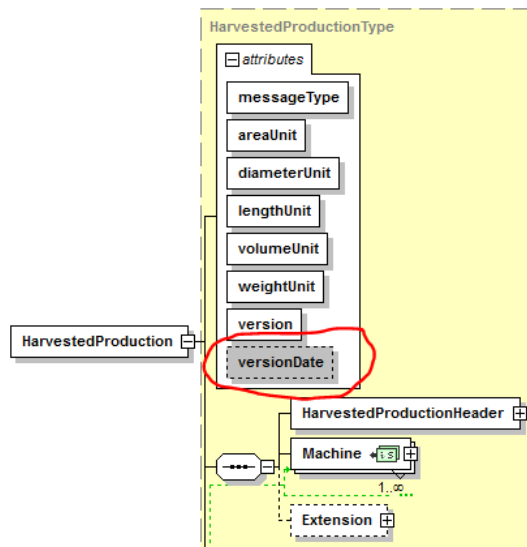
It was also suggested to add some date stamp to the schemas in order to improve management of primarily the draft and pre-release schemas. The main reason for this is that both draft and pre-release versions might be modified more than once.

A new attribute versionDate has therefore been included in two different locations.

A timestamp has been added to attributes of schema headers:

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns="urn:skogforsk:stanford2010"
xmlns:doc="urn:skogforsk:stanford2010:doc"
xmlns:sfd="urn:skogforsk:stanford2010"
targetNamespace="urn:skogforsk:stanford2010"
elementFormDefault="qualified"
attributeFormDefault="unqualified"
version="release_2.1"
sfd:versionDate="2012-09-06">
```

An optional attribute versionDate has been added to attribute list of each message in version 2.1 (will be mandatory in version 3.0):

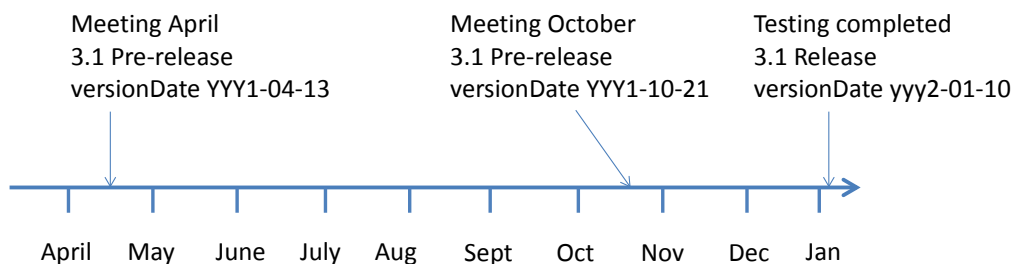


Roadmap

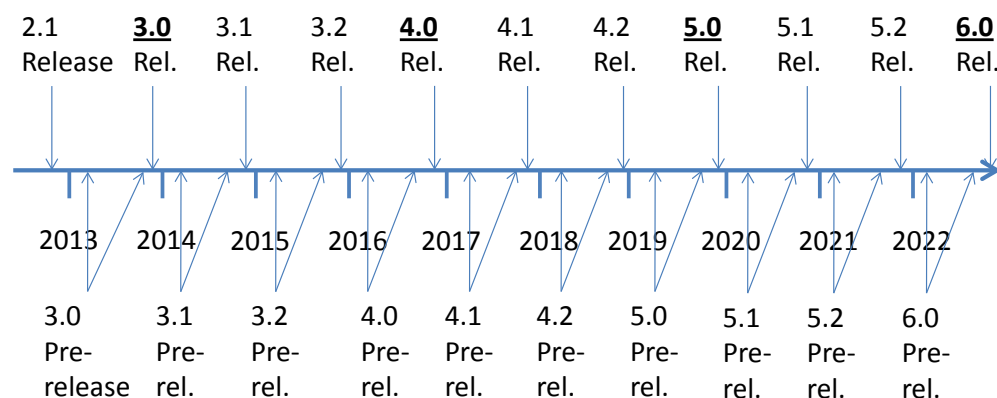
The basic rules for updating StanForD2010 will be:

- Never more than one new "release" (minor or major) per year.
- Maximum number of major releases is once per three years.
- Two meetings: March and October
- All decisions at meetings are included in "pre-release". No other changes allowed in "pre-release".
- A "pre-release" will become a "release" after a two/three month testing period.

The figure below illustrates the normal updating of StanForD2010 during a year assuming that we use status type pre-release and versionDate attribute.



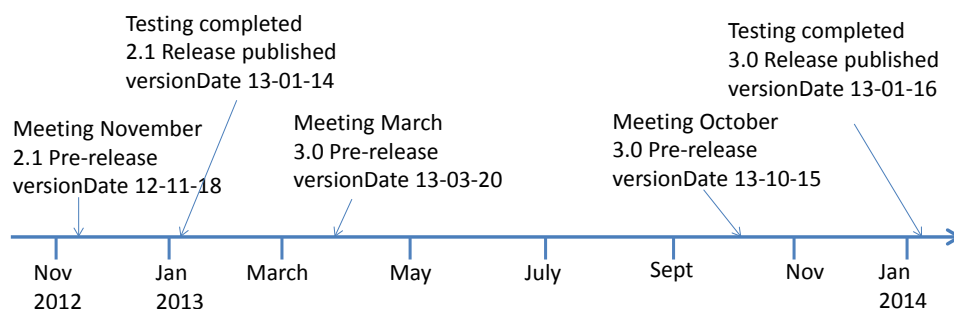
The rules can also be illustrated by the following figure below.



New major 3.0 during 2013

Based on the rules and roadmap above as well as the fact that there are also additional suggested “major” modifications (e.g. in foi) it was decided to handle the issue of updating to 3.0 according to the following text:

1. A new 2.1 pre-release is published after meeting in November and a release is published in January 2013 after a testing period.
2. A new 3.0 pre-release is published after meeting in March.
3. Possible new additional updates to 3.0 pre-release after meeting in October.
4. ”Pre-release” will become a “release” after a two/three month testing period.





John Arlinger

11 October 2012

User defined data within StanForD2010

Background

The proposals in this document are originally based on suggestions from Peter Assarsson, Komatsu.

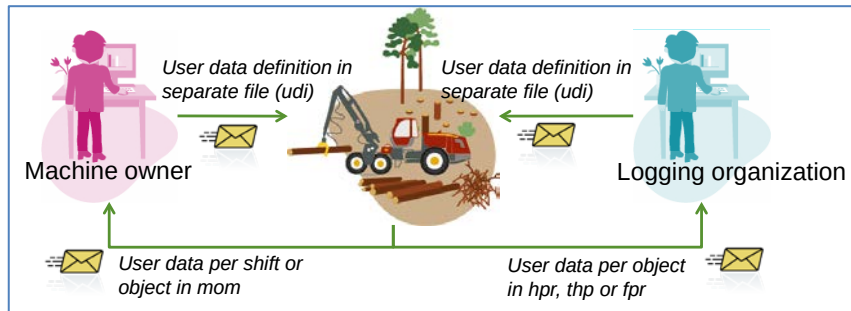
A new message called invoice specification used for registering extra work and material was discussed during 2011. The main objective of this message was to make it possible to register events inside or outside forest machines that are to be included when machine owner is invoicing logging organization. Today this is often solved either by having the operator writing on a piece of paper or in an excel-sheet. A large number of other examples where excel-sheets are used for registering relevant information in forest machines could probably be found out in the “real world”.

A problem with standardizing this type of information is that the needs probably vary significantly between different organizations and markets. An example might be trying to compile a list of different work categories. Below follows a possible list of such extra work categories:

Harvesting	MachinePlanting	MachineTransfer
SpecialHarvesting	YoungStandTreatment	RoadMaintenance
ManualHarvesting	StandInventory	ControlMeasurement
Forwarding	UndergrowthCleaning	WorkSitePlanning
ExtendedForwarding	StumpTreatment	ControlOfWorkQuality
SpecialForwarding	ProcessingOfLoggingResidues	OtherMachineWork
SitePreparation	CoveringOfEnergyWood	OtherManualWork

It would be an advantage if we could find a flexible solution for both sending instructions and reporting user defined data from forest machines.

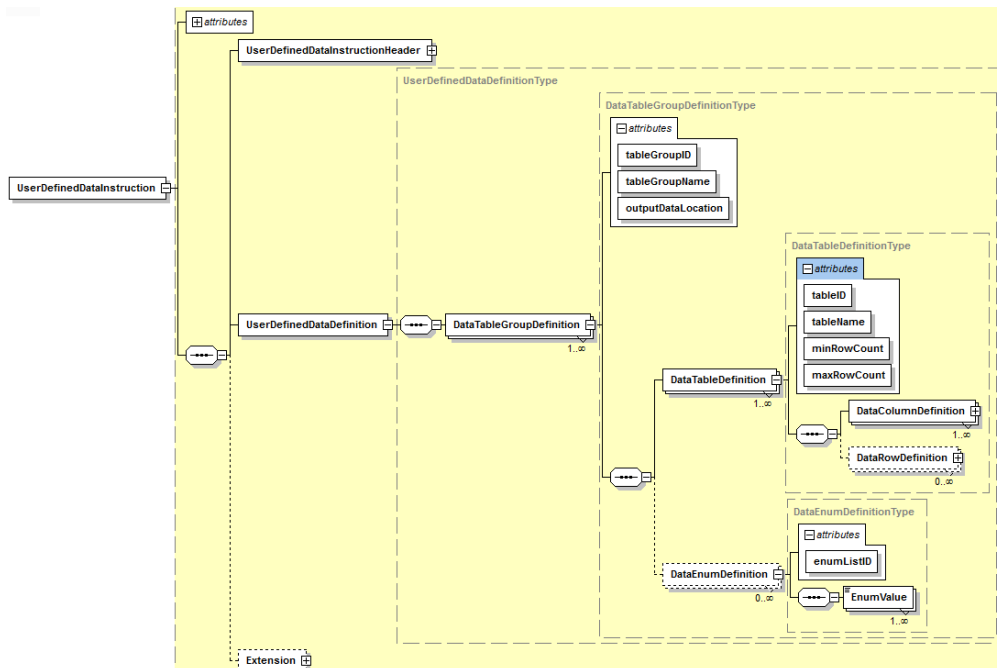
The figure below illustrates a possible solution for how instructions could be sent to a machine from either forest machine owner or logging organization and then returned together within a StanForD message.



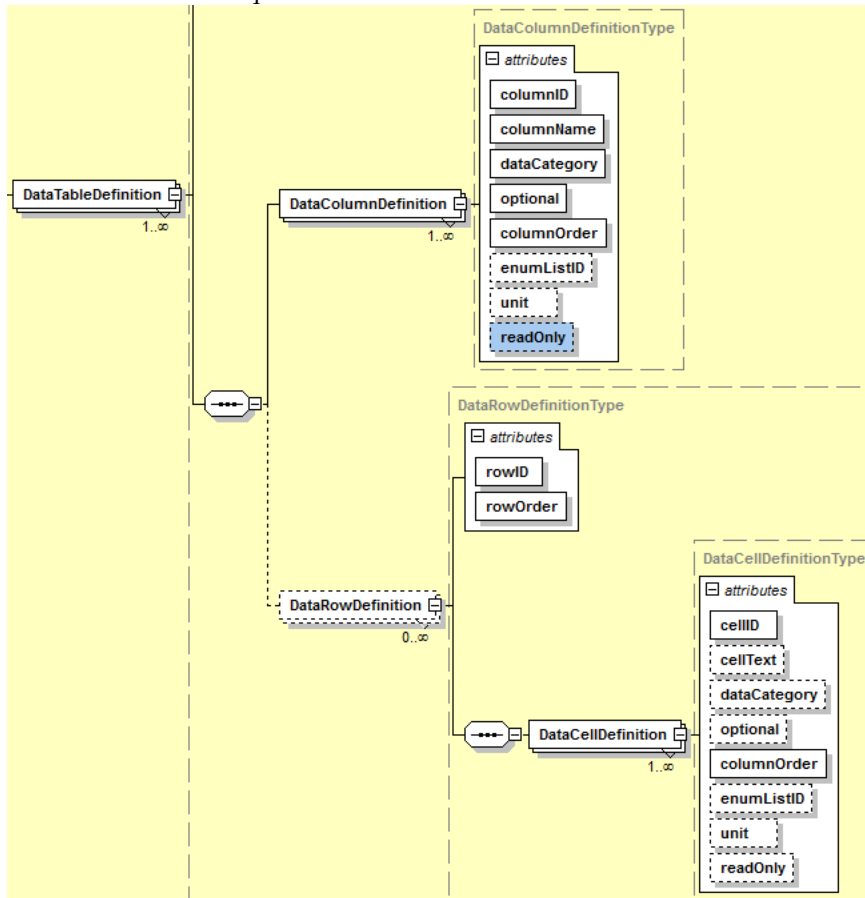
The basic idea is to make it possible to define tables that the operator can use in the machine to manually register data relevant for logging organization or machine owner.

Input / Instruction

Below is a suggested schema describing what the structure of an instruction should look like:



Observe that the DataRowDefinition was added to make it possible to pre-define a number of questions with different enumeration lists for different rows.



DataTableGroupDefinition.

This structure is comparable with a database or an “excel work book” to be reported in a single way.

Attributes for DataTableGroupDefinition:

- tableGroupId (mandatory, string)
- tableGroupName (mandatory, string) = Name of user defined data table, to be used in the UI
- outputDataLocation (mandatory, enumeration list)

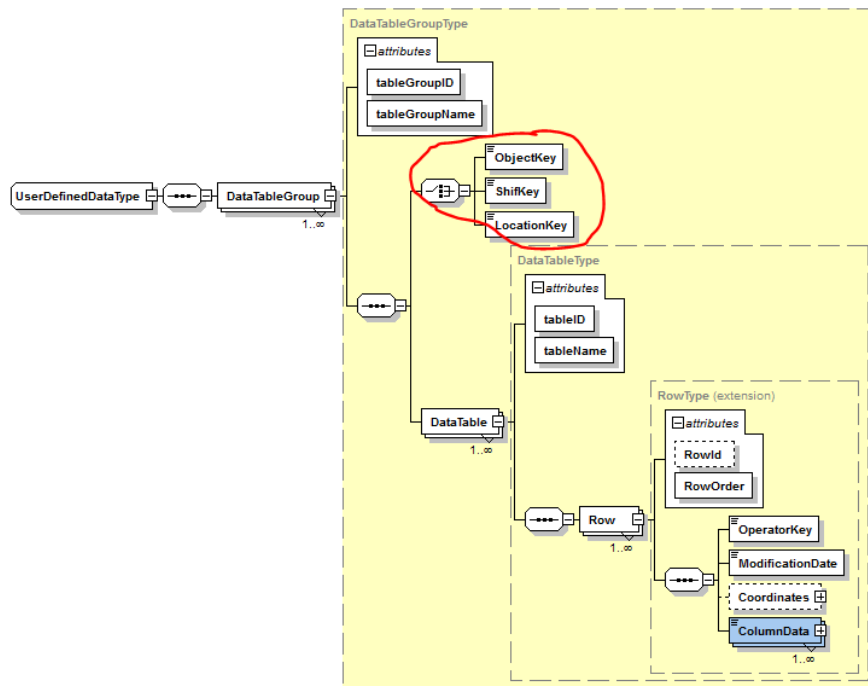
DataTableGroupDefinition is repeatable. The machine owner might need information both per shift and per harvesting object. This requires that there are possibilities to add many DataTableGroupDefinitions. The idea is that for every DataTableGroupDefinition there will be a new "tab" in the dialog.

outputDataLocation

Attribute ***outputDataLocation*** points to where the user defined data should be registered and then reported. The following enumerations are suggested:

- *ProductionObject*, information is stored per object in hpr, thp and fpr files (ObjectKey in illustration below is used).
- *MonitoringObject*, information is stored per object in mom files (ObjectKey in illustration below is used).
- *MonitoringShift*, information is stored per shift in mom files (ShiftKey in illustration below is used)). This means that the element ShiftKey must be used.
- *ProductionLocation*, information is stored per location in fpr files (LocationKey in illustration below is used). A recommendation could be to assume that the LocationKey references the last location where any volumes have been unloaded if several different locations are used at the same object.

The output data is stored using the following structure in hpr, thp, fpr and mom.



Tables with ProductionObject and MonitoringObject are object specific tables that are to be reset with a new object. If outputLocation is equal to MonitoringShift the table is shift specific, meaning it is to be reset when starting on new shift.

DataTableDefinition.

This structure is comparable with a database table or an “excel work sheet”.

Attributes for DataTableDefinition:

- tableId (mandatory, string)
- tableName (mandatory, string) = Name of user defined data table, to be used in the UI
- maxRowCount (mandatory, ”1” or “maxint”) = Maximum number of rows in user defined data table.
- minRowCount (mandatory, ”0” or “1”) = Minimum number of rows in user defined data table.

DataColumnDefinition

This structure is comparable with fields in a database table or columns in an “excel-sheet”.

Attributes for DataColumnDefinition:

- columnId (mandatory, string) = Identity of column
- columnName (mandatory, string) = Name of column. Presented in column header.
- dataCategory (mandatory, enumerations =”enum”, “string”, “integer”, “decimal”, “date”, “boolean” or “sequentialInteger”): Defines data type of column.

- optional (mandatory, boolean) = Indicates if column must be filled in by operator.
- columnOrder (mandatory, integer)= Presentation order of column in GUI.
- enumListId (optional, string)= Identity of enumeration list. E.g. "UnitList", refers to a DataEnumDefinition (see also next section)
- unit (optional, string) = Defines unit of column, eg mm, kg, m3, cm etc. Can be used in GUI to inform operator about unit.

DataRowDefinition

This new suggested structure is used in order to define individual cells within a row if there e.g. is a need for defining different enumerations on different rows. All included setting must override settings in DataColumnDefinition.

Attributes for DataRowDefinition:

- rowId (mandatory, string) = Identity of row
- rowOrder (mandatory, integer) = Presentation order of row in GUI.

Attributes for DataCellDefinition:

- cellId (mandatory, string) = Identity of cell
- cellText (optional, string) = Text in cell. E.g. a question to be answered by operator.
- dataCategory (optional, enumerations ="enum", "string", "integer", "decimal", "date", "boolean" or "sequentialInteger") = Defines data type of current cell, overrides DataColumnDefinition. Allowed values are: enum, integer, string, decimal, date, boolean, sequential no per row.
- optional (optional, boolean): Indicates if cell must be filled in by operator.
- columnOrder (mandatory, integer) = Column order for current cell in GUI.
- enumListId (optional, string) = Identity of enumeration list. E.g. "UnitList", refers to a DataEnumDefinition (see also next section)
- unit (optional, string) = Defines unit of column, eg mm, kg, m3, cm etc. Can be used in GUI to inform operator about unit.
- readOnly (optional, Boolean) = Indicates if cell can be edited by operator.

DataEnumDefinition

This structure includes enumeration lists to be used in the machine. Attribute enumListId identifies the enumeration list to be used.

XML example "Invoice specification"

Example of xml-file with instruction for how to report user defined data (observe that columnOrder is excluded in order to decrease size):

```
<UserDefinedDataInstruction diameterUnit="mm" messageType="udi" volumeUnit="m3" areaUnit="ha" lengthUnit="cm" version="1.0" weightUnit="kg"
xsi:schemaLocation="urn:skogforsk:stanford2010 UserDefinedDataInstruction_V2p1_120907.xsd" xmlns="urn:skogforsk:stanford2010"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <UserDefinedDataInstructionHeader>
```

```

<SenderApplication>String</SenderApplication>
<CreationDate>2011-09-23T10:27:48+02:00</CreationDate>
<ModificationDate>2011-09-23T10:27:48+02:00</ModificationDate>
<ApplicationVersionCreated>String</ApplicationVersionCreated>
<ApplicationVersionModified>String</ApplicationVersionModified>
<CountryCode>String</CountryCode>
</UserDefinedDataInstructionHeader>
<UserDefinedDataDefinition>
<DataTableGroupDefinition outputDataLocation="MonitoringObject" tableGroupID="Basics_for_invoicing" tableGroupName="Basics for invoicing">
  <DataEnumDefinition enumListID="BasicsList">
    <EnumValue>Ordered by forest company</EnumValue>
    <EnumValue>Ordered by forest owner</EnumValue>
  </DataEnumDefinition>
  <DataEnumDefinition enumListID="ExtraList">
    <EnumValue>Harvesting (normal harvesting)</EnumValue>
    <EnumValue>SpecialHarvesting</EnumValue>
    <EnumValue>ManualHarvesting</EnumValue>
    <EnumValue>Forwarding (normal forwarding)</EnumValue>
    <EnumValue>ExtendedForwarding</EnumValue>
    <EnumValue>SpecialForwarding</EnumValue>
    <EnumValue>SitePreparation, divided according to method</EnumValue>
    <EnumValue>MachinePlanting</EnumValue>
    <EnumValue>YoungStandTreatment pre-commercial thinning </EnumValue>
    <EnumValue>StandInventory (manual measurements)</EnumValue>
    <EnumValue>UndergrowthCleaning</EnumValue>
    <EnumValue>StumpTreatment</EnumValue>
    <EnumValue>ProcessingOfLoggingResidues</EnumValue>
    <EnumValue>CoveringOfEnergyWood</EnumValue>
    <EnumValue>MachineTransfer</EnumValue>
    <EnumValue>RoadMaintenance</EnumValue>
    <EnumValue>ControlMeasurement</EnumValue>
    <EnumValue>WorkSitePlanning</EnumValue>
    <EnumValue>ControlOfWorkQuality</EnumValue>
    <EnumValue>OtherMachineWork</EnumValue>
    <EnumValue>OtherManualWork</EnumValue>
  </DataEnumDefinition>
  <DataEnumDefinition enumListID="UnitList">
    <EnumValue>kr</EnumValue>
    <EnumValue>Kg</EnumValue>
    <EnumValue>Litre</EnumValue>
    <EnumValue>m3</EnumValue>
    <EnumValue>ha</EnumValue>
  </DataEnumDefinition>
  <DataEnumDefinition enumListID="ExtraWorkMaterialList">
    <EnumValue>Beer</EnumValue>
    <EnumValue>Whiskey</EnumValue>
    <EnumValue>Paper</EnumValue>
    <EnumValue>Fungicide</EnumValue>
    <EnumValue>Paint</EnumValue>
  </DataEnumDefinition>
  <DataTableDefinition tableName="Customer" tableID="Customer" maxRowCount="1" minRowCount="1">
    <DataColumnDefinition columnName="Customer id" columnID="Customer id" dataCategory="integer" optional="false"/>
    <DataColumnDefinition columnName="Name" columnID="Name" dataCategory="string" optional="true"/>
  </DataTableDefinition>
  <DataTableDefinition tableName="Extra work" tableID="Extra work" maxRowCount="99999" minRowCount="0">
    <DataColumnDefinition columnName="Type of work" columnID="TypeWork" dataCategory="enum" optional="false" enumListID="ExtraList"/>
    <DataColumnDefinition columnName="Basic" columnID="BasicInvoic" dataCategory="enum" optional="false" enumListID="BasicsList"/>
    <DataColumnDefinition columnName="Quantity" columnID="Quantity" dataCategory="integer" optional="false"/>
    <DataColumnDefinition columnName="Unit" columnID="Unit" dataCategory="enum" optional="false" enumListID="UnitList"/>
    <DataColumnDefinition columnName="Comment" columnID="Comment" dataCategory="string" optional="true"/>
  </DataTableDefinition>
  <DataTableDefinition tableName="Invoiced material" tableID="Invoiced material" maxRowCount="99999" minRowCount="0">
    <DataColumnDefinition columnName="Type" columnID="Type" dataCategory="enum" optional="false" enumListID="ExtraWorkMaterialList"/>
    <DataColumnDefinition columnName="Quantity" columnID="Quantity" dataCategory="integer" optional="false"/>
    <DataColumnDefinition columnName="Unit" columnID="Unit" dataCategory="enum" optional="false" enumListID="UnitList"/>
    <DataColumnDefinition columnName="Comment" columnID="Comment" dataCategory="string" optional="true"/>
  </DataTableDefinition>
</DataTableGroupDefinition>
<DataTableGroupDefinition outputDataLocation="Production" tableGroupID="Follow-up harvesting">
  <DataEnumDefinition enumListID="Species">
    <EnumValue>Pine</EnumValue>
    <EnumValue>Spruce</EnumValue>
    <EnumValue>Other broadleaves</EnumValue>
    <EnumValue>Birch</EnumValue>
  </DataEnumDefinition>
  <DataTableDefinition tableName="High stumps" tableID="High stumps" maxRowCount="99999" minRowCount="0">
    <DataColumnDefinition columnName="Species" columnID="Species" dataCategory="enum" optional="false" enumListID="Species"/>

```

```

    <DataColumnDefinition columnName="NoOfStems" columnID="NoOfStems" dataCategory="integer" optional="false"/>
    <DataColumnDefinition columnName="Coordinate" columnID="Coordinate" dataCategory="string" optional="true"/>
  </DataTableDefinition>
  <DataTableDefinition tableName="Eternal trees" tableID="Eternal trees" maxRowCount="99999" minRowCount="0">
    <DataColumnDefinition columnName="Species" columnID="Species" dataCategory="enum" optional="false" enumListID="Species"/>
    <DataColumnDefinition columnName="NoOfStems" columnID="NoOfStems" dataCategory="integer" optional="false"/>
    <DataColumnDefinition columnName="Coordinate" columnID="Coordinate" dataCategory="string" optional="true"/>
  </DataTableDefinition>
</DataTableGroupDefinition>
</UserDefinedDataDefinition>
</UserDefinedDataInstruction>

```

User interface example

The instruction described above could render the following GUI.

First table:

InvoiceSpecification		
InvoicedParty	ExtraWorkEvent	InvoicedMaterial
OperatorKey / Date	CustomerId	Name

Second table:

InvoiceSpecification					
InvoicedParty	ExtraWorkEvent	InvoicedMaterial			
OperatorKey / Date	ExtraWorkCategory	BasicsForInvoicingTheWork	Quantity	Unit	Comment

Third table:

InvoiceSpecification				
InvoicedParty	ExtraWorkEvent	InvoicedMaterial		
OperatorKey / Date	Type	Quantity	Unit	Comment

Enumeration list for column “Type” in third table:

y	ExtraWorkEvent	InvoicedMaterial	
/ / Date	Type	Quantity	U
11/10/2010 10:36:44.23	ExtendedForwarding		
	ExtendedForwarding		
	SpecialForwarding		
	SitePreparation, divided ac		
	MachinePlanting		
	YoungStandTreatment, pr		
	StandInventory (manual me		
	UndergrowthCleaning		
	StumpTreatment		
	ProcessingOfLoggingResid		
	CoveringOfEnergyWood		
	MachineTransfer		
	RoadMaintenance		
	ControlMeasurement		

XML example “questionnaire” per object

Using the new optional structure RowDataStructure it is also possible to pre-define a number of questions with different enumeration lists for different rows (header excluded from example):

```
<
<DataTableDefinition tableName="Following up questionnaire" tableID="FollowUpForm" maxRowCount="7" minRowCount="7">
  <DataColumnDefinition columnName="Question type" columnID="QT" dataCategory="string" optional="false" columnOrder="1"></DataColumnDefinition>
  <DataColumnDefinition columnName="Question" columnID="Q1" dataCategory="string" optional="false" columnOrder="2"></DataColumnDefinition>
  <DataColumnDefinition columnName="Answer" columnID="Answ1" dataCategory="string" optional="true" columnOrder="3"></DataColumnDefinition>
  <DataColumnDefinition columnName="Comment" columnID="Com1" dataCategory="string" optional="false" columnOrder="4"></DataColumnDefinition>
  <DataRowDefinition rowID="r1" rowOrder="1">
    <DataCellDefinition cellID="r1c1" cellText="Question 1a, follow up" columnOrder="1" dataCategory="string" readOnly="true"/>
    <DataCellDefinition cellID="r1c2" cellText="Creeks and wetlands undamaged?" columnOrder="2" dataCategory="string" readOnly="true"/>
    <DataCellDefinition cellID="r1c3" columnOrder="3" dataCategory="boolean" optional="false" readOnly="false"/>
    <DataCellDefinition cellID="r1c4" columnOrder="4" dataCategory="string" optional="true" readOnly="false"/>
  </DataRowDefinition>
  <DataRowDefinition rowID="r2" rowOrder="2">
    <DataCellDefinition cellID="r2c1" cellText="Question 1b, follow up" columnOrder="1" dataCategory="string" readOnly="true"/>
    <DataCellDefinition cellID="r2c2" cellText="Protection areas intact?" columnOrder="2" dataCategory="string" readOnly="true"/>
  </DataRowDefinition>

```

```

    <DataCellDefinition cellID="r2c3" columnOrder="3" dataCategory="boolean" optional="false" readOnly="false"/>
    <DataCellDefinition cellID="r2c4" columnOrder="4" dataCategory="string" optional="true" readOnly="false"/>
  </DataRowDefinition>
  <DataRowDefinition rowID="r3" rowOrder="3">
    <DataCellDefinition cellID="r3c1" cellText="Question 2a deviations" columnOrder="1" dataCategory="string" readOnly="true"/>
    <DataCellDefinition cellID="r3c2" cellText="Deviations (when and why)?" columnOrder="2" dataCategory="string" readOnly="true"/>
    <DataCellDefinition cellID="r3c3" columnOrder="3" dataCategory="date" optional="false" readOnly="false"/>
    <DataCellDefinition cellID="r3c4" columnOrder="4" dataCategory="string" optional="true" readOnly="false"/>
  </DataRowDefinition>
  <DataRowDefinition rowID="r4" rowOrder="4">
    <DataCellDefinition cellID="r4c1" cellText="Question 2b deviations" columnOrder="1" dataCategory="string" readOnly="true"/>
    <DataCellDefinition cellID="r4c2" cellText="Corrected actions (when and why)?" columnOrder="2" dataCategory="string" readOnly="true"/>
    <DataCellDefinition cellID="r4c3" columnOrder="3" dataCategory="date" optional="false" readOnly="false"/>
    <DataCellDefinition cellID="r4c4" columnOrder="4" dataCategory="string" optional="true" readOnly="false"/>
  </DataRowDefinition>
  <DataRowDefinition rowID="r5" rowOrder="5">
    <DataCellDefinition cellID="r5c1" cellText="Question 3a landing preparation" columnOrder="1" dataCategory="string" readOnly="true"/>
    <DataCellDefinition cellID="r5c2" cellText="Road fixed?" columnOrder="2" dataCategory="string" readOnly="true"/>
    <DataCellDefinition cellID="r5c3" columnOrder="3" dataCategory="enum" enumListID="GoodBad" optional="false" readOnly="false"/>
    <DataCellDefinition cellID="r5c4" columnOrder="4" dataCategory="string" optional="true" readOnly="false"/>
  </DataRowDefinition>
  <DataRowDefinition rowID="r6" rowOrder="6">
    <DataCellDefinition cellID="r6c1" cellText="Question 3b landing preparation" columnOrder="1" dataCategory="string" readOnly="true"/>
    <DataCellDefinition cellID="r6c2" cellText="Landing fixed?" columnOrder="2" dataCategory="string" readOnly="true"/>
    <DataCellDefinition cellID="r6c3" columnOrder="3" dataCategory="enum" enumListID="GoodBad" optional="false" readOnly="false"/>
    <DataCellDefinition cellID="r6c4" columnOrder="4" dataCategory="string" optional="true" readOnly="false"/>
  </DataRowDefinition>
  <DataRowDefinition rowID="r7" rowOrder="7">
    <DataCellDefinition cellID="r7c1" cellText="Question 3c landing preparation" columnOrder="1" dataCategory="string" readOnly="true"/>
    <DataCellDefinition cellID="r7c2" cellText="Turning place for truck (24 m)?" columnOrder="2" dataCategory="string" readOnly="true"/>
    <DataCellDefinition cellID="r7c3" columnOrder="3" dataCategory="enum" enumListID="GoodBad" optional="false" readOnly="false"/>
    <DataCellDefinition cellID="r7c4" columnOrder="4" dataCategory="string" optional="true" readOnly="false"/>
  </DataRowDefinition>
</DataTableDefinition>
<DataEnumDefinition enumListID="Landing">
  <EnumValue>OK, accessible for transport</EnumValue>
  <EnumValue>Landin full</EnumValue>
  <EnumValue>Not accessible for transport</EnumValue>
</DataEnumDefinition>
<DataEnumDefinition enumListID="Road">
  <EnumValue>Snow, must be plowed</EnumValue>
  <EnumValue>Gravel needed</EnumValue>
</DataEnumDefinition>
<DataEnumDefinition enumListID="YesNo">
  <EnumValue>Yes</EnumValue>
  <EnumValue>No</EnumValue>
</DataEnumDefinition>
<DataEnumDefinition enumListID="GoodBad">
  <EnumValue>Good</EnumValue>
  <EnumValue>Ok</EnumValue>
  <EnumValue>Bad</EnumValue>
</DataEnumDefinition>
</DataTableDefinition>

```

User interface example

The instruction described above could render the following GUI (example is filled in by operator):

Follow-up forwarding (Production)				
Landing Following up questionnaire				
OperatorKey / Date	Question type	Question	Answer	Comment
3 / 2012-05-29T12:35:17.96	Question 1a, follow up	Creeks and wetlands undamaged?	TRUE	
3 / 2012-05-29T12:35:24.61	Question 1b, follow up	Protection areas intact?	FALSE	
3 / 2012-05-29T12:37:16.99	Question 2a deviations	Deviations (when and why)?	2012-05-15T	Oil spill
3 / 2012-05-29T12:36:47.77	Question 2b deviations	Corrected actions (when and why)?	2012-05-16T	Cleaned up oil spill
3 / 2012-05-29T12:36:10.36	Question 3a landing preparation	Road fixed?	Ok	
3 / 2012-05-29T12:36:10.83	Question 3b landing preparation	Landing fixed?	Good	
3 / 2012-05-29T12:36:37.10	Question 3c landing preparation	Turning place for truck (24 m)?	Bad	Re-loading need
			Good	
			Ok	
			Bad	

XML example "questionnaire" per location

Using the new optional structure RowDataStructure it is also possible to pre-define a number of questions with different enumeration lists for different rows (header excluded from example):

```
<DataTableGroupDefinition outputDataLocation="ProductionLocation" tableGroupID="Follow-up forwarding" tableGroupName="Landing data">
  <DataTableDefinition tableName="Status information" tableID="StatusForm" maxRowCount="3" minRowCount="3">
    <DataColumnDefinition columnName="Question" columnID="Q1" dataCategory="string" optional="false" columnOrder="1"></DataColumnDefinition>
    <DataColumnDefinition columnName="Answer" columnID="Answ1" dataCategory="string" optional="true" columnOrder="2"></DataColumnDefinition>
    <DataColumnDefinition columnName="Comment" columnID="Cm1" dataCategory="string" optional="false" columnOrder="3"></DataColumnDefinition>
    <DataRowDefinition rowID="r1" rowOrder="1">
      <DataCellDefinition cellID="r1c1" cellText="Road status?" columnOrder="1" dataCategory="string" readOnly="true"/>
      <DataCellDefinition cellID="r1c2" cellText="Ok" columnOrder="2" dataCategory="enum" enumListID="Road" optional="false" readOnly="false"/>
      <DataCellDefinition cellID="r1c3" columnOrder="3" dataCategory="string" optional="true" readOnly="false"/>
    </DataRowDefinition>
    <DataRowDefinition rowID="r2" rowOrder="2">
      <DataCellDefinition cellID="r2c1" cellText="Landing status?" columnOrder="1" dataCategory="string" readOnly="true"/>
      <DataCellDefinition cellID="r2c2" cellText="Ok" columnOrder="2" dataCategory="enum" enumListID="Landing" optional="false" readOnly="false"/>
      <DataCellDefinition cellID="r2c3" columnOrder="3" dataCategory="string" optional="true" readOnly="false"/>
    </DataRowDefinition>
    <DataRowDefinition rowID="r3" rowOrder="3">
      <DataCellDefinition cellID="r3c1" cellText="Turning place for truck (24 m)?" columnOrder="1" dataCategory="string" readOnly="true"/>
      <DataCellDefinition cellID="r3c2" cellText="Ok" columnOrder="2" dataCategory="enum" enumListID="YesNo" optional="false" readOnly="false"/>
      <DataCellDefinition cellID="r3c3" columnOrder="3" dataCategory="string" optional="true" readOnly="false"/>
    </DataRowDefinition>
  </DataTableDefinition>
  <DataEnumDefinition enumListID="Landing">
    <EnumValue>OK, accessible for transport</EnumValue>
    <EnumValue>Landing full, truck transportation need</EnumValue>
    <EnumValue>Not accessible for transport</EnumValue>
  </DataEnumDefinition>
  <DataEnumDefinition enumListID="Road">
    <EnumValue>Ok</EnumValue>
    <EnumValue>Snow, must be plowed</EnumValue>
    <EnumValue>Gravel needed</EnumValue>
  </DataEnumDefinition>
  <DataEnumDefinition enumListID="YesNo">
    <EnumValue>Yes</EnumValue>
    <EnumValue>No</EnumValue>
  </DataEnumDefinition>
</DataTableGroupDefinition>
```

User interface example

The instruction described above could render the following GUI (example is filled in by operator):

Landing data (ProductionLocation)			
Status information			
OperatorKey / Date	Question	Answer	Comment
3 / 2012-10-11T14:45:32.29	Road status?	Gravel needed	
3 / 2012-10-11T14:45:47.09	Landing status?	Not accessible for transport	
	Turning place for truck (24 m)?	No Yes No	

XML example “questionnaire” per shift

```
<DataTableGroupDefinition outputDataLocation="MonitoringShift" tableGroupID="SFU" tableGroupName="Shift follow-up">
  <DataTableDefinition tableName="Consumption" tableID="Förbrukning" maxRowCount="2" minRowCount="2">
    <DataColumnDefinition columnName="Material" columnID="Mtrl" dataCategory="string" optional="false" columnOrder="1"/>
    <DataColumnDefinition columnName="Amount (l)" columnID="Q10" dataCategory="decimal" optional="false" columnOrder="2"/>
    <DataRowDefinition rowID="r1" rowOrder="1">
      <DataCellDefinition cellID="r1c1" cellText="Saw oil?" columnOrder="1" dataCategory="string" readOnly="true"/>
    </DataRowDefinition>
    <DataRowDefinition rowID="r2" rowOrder="2">
      <DataCellDefinition cellID="r2c1" cellText="Hydraulic oil?" columnOrder="1" dataCategory="string" readOnly="true"/>
    </DataRowDefinition>
  </DataTableDefinition>
  <DataTableDefinition tableName="Travel expences" tableID="TE" maxRowCount="1" minRowCount="1">
    <DataColumnDefinition columnName="Typ" columnID="Mtrl" dataCategory="string" optional="false" columnOrder="1"/>
    <DataColumnDefinition columnName="Amount" columnID="Q11" dataCategory="integer" optional="false" columnOrder="2"/>
    <DataColumnDefinition columnName="Unit" columnID="U11" dataCategory="string" optional="false" columnOrder="2"/>
    <DataRowDefinition rowID="r1" rowOrder="1">
      <DataCellDefinition cellID="r1c1" cellText="Distance to work?" columnOrder="1" dataCategory="string" readOnly="true"/>
      <DataCellDefinition cellID="r1c1" cellText="km" columnOrder="3" dataCategory="string" readOnly="true"/>
    </DataRowDefinition>
  </DataTableDefinition>
</DataTableGroupDefinition>
```

User interface example

The instruction described above could render the following GUI:

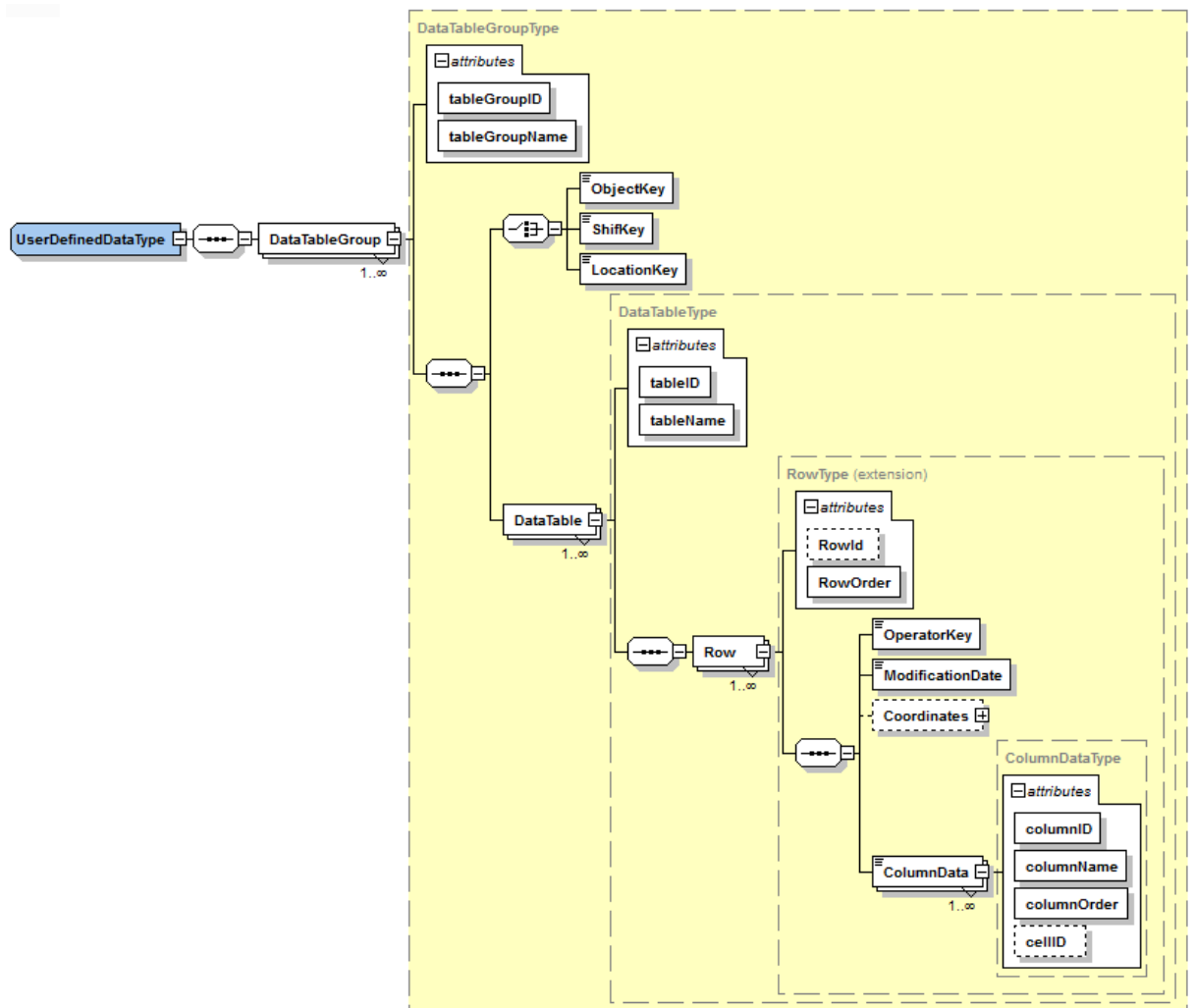
Follow-up forwarding (ProductionObject)		Shift follow-up (MonitoringShift)	
Consumption		Travel expences	
OperatorKey / Date	Material	Amount (l)	
	Saw oil?		
	Hydraulic oil?		

Follow-up forwarding (ProductionObject)		Shift follow-up (MonitoringShift)	
Consumption		Travel expences	
OperatorKey / Date	Typ	Amount	Unit
	Distance to work?		km

Output / Report

The data is validated according to the definition before the file is created. This means that all mandatory fields have to be filled in.

Below is a suggested schema structure for reporting user defined data from the machine, always included under Machine in mom, fpr, hpr or thp:



XML example

This is an example of the output file (all production data and some machine, object and operator information is excluded):

```
<UserDefinedDataMessage xsi:schemaLocation="urn:skogforsk:stanford2010:1 TotalHarvestedProduction_V2p1_120907.xsd"
xmlns="urn:skogforsk:stanford2010" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">

  <HarvestedProduction diameterUnit="mm" messageType="hpr" volumeUnit="m3" areaUnit="ha" lengthUnit="cm" version="2.1"
weightUnit="kg" xsi:schemaLocation="urn:skogforsk:stanford2010 HarvestedProduction_V2p1.xsd"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:sfd="urn:skogforsk:stanford2010">
    <HarvestedProductionHeader>
      <CreationDate>2011-09-23T10:27:48+02:00</CreationDate>
      <ModificationDate>2011-10-04T15:01:50.080+02:00</ModificationDate>
    </HarvestedProductionHeader>
    <MachineDefinition machineCategory="Harvester">
      <MachineKey>12389</MachineKey>
      <MachineUserID>Machine 121</MachineUserID>
      <OperatorDefinition>
        <OperatorKey>2</OperatorKey>
        <OperatorUserID>String</OperatorUserID>
        <ContactInformation>
          <FirstName>Daniel</FirstName>
          <LastName>Smith</LastName>
        </ContactInformation>
      </OperatorDefinition>
    </MachineDefinition>
    <ObjectDef>
      <ObjectUserID agency="LoggingOrganization">String</ObjectUserID>
      <ObjectName>String</ObjectName>
      <ContractNumber ContractCategory="String">String</ContractNumber>
      <RealEstateIDObject agency="LoggingOrganization">String</RealEstateIDObject>
      <ObjectKey>54535</ObjectKey>
    </ObjectDef>
  </HarvestedProduction>
</UserDefinedDataMessage>
```

```

    <StartDate>2011-09-23T10:27:48+02:00</StartDate>
  </ObjectDef>
  <DataTableGroup tableName="InvoiceSpecification" tableId="InvoiceSpecification">
    <ObjectKey>54535</ObjectKey>
    <DataTable tableId="InvoicedParty">
      <Row rowOrder="1">
        <OperatorKey>3</OperatorKey>
        <ModificationDate>2001-12-17T09:30:47Z</ModificationDate>
        <ColumnData columnName="CustomerId" columnId="C" columnOrder="1">35643</ColumnData>
        <ColumnData columnName="Name" columnId="N" columnOrder="2">BigForest Lmtd</ColumnData>
      </Row>
    </DataTable>
    <DataTable tableName="ExtraWorkEvent" tableId="ExtraWorkEvent">
      <Row rowOrder="1">
        <OperatorKey>3</OperatorKey>
        <ModificationDate>2001-12-17T09:30:47Z</ModificationDate>
        <ColumnData columnName="ExtraWorkCategory" columnId="E"
columnOrder="1">YoungStand Treatment</ColumnData>
        <ColumnData columnName="BasicsForInvoicingTheWork" columnId="B" columnOrder="1">Ordered by forest owner
</ColumnData>
        <ColumnData columnName="Quantity" columnId="Q" columnOrder="2">26</ColumnData>
        <ColumnData columnName="Unit" columnId="U" columnOrder="3">hours</ColumnData>
        <ColumnData columnName="Comment" columnId="C" columnOrder="4">This is a test of invoice extrawork in
UserDefinedData</ColumnData>
      </Row>
      <Row rowOrder="2">
        <OperatorKey>3</OperatorKey>
        <ModificationDate>2001-12-17T09:30:47Z</ModificationDate>
        <ColumnData columnName="ExtraWorkCategory" columnId="E" columnOrder="1">StandInventory</ColumnData>
        <ColumnData columnName="BasicsForInvoicingTheWork" columnId="B" columnOrder="2">Ordered by logging
company</ColumnData>
        <ColumnData columnName="Quantity" columnId="Q" columnOrder="3">4</ColumnData>
        <ColumnData columnName="Unit" columnId="U" columnOrder="4">hours</ColumnData>
      </Row>
      <Row rowOrder="3">
        <OperatorKey>3</OperatorKey>
        <ModificationDate>2001-12-17T09:30:47Z</ModificationDate>
        <ColumnData columnName="ExtraWorkCategory" columnId="E" columnOrder="1">StandInventory</ColumnData>
        <ColumnData columnName="BasicsForInvoicingTheWork" columnId="B" columnOrder="2">Ordered by logging
company</ColumnData>
        <ColumnData columnName="Quantity" columnId="Q" columnOrder="3">4.5</ColumnData>
        <ColumnData columnName="Unit" columnId="U" columnOrder="4">ha</ColumnData>
      </Row>
    </DataTable>
    <DataTable tableName="InvoicedMaterial" tableId="InvoicedMaterial">
      <Row rowOrder="1">
        <OperatorKey>3</OperatorKey>
        <ModificationDate>2001-12-17T09:30:47Z</ModificationDate>
        <ColumnData columnName="Type" columnId="T" columnOrder="1">ManualHarvesting</ColumnData>
        <ColumnData columnName="Quantity" columnId="Q" columnOrder="2">60</ColumnData>
        <ColumnData columnName="Unit" columnId="U" columnOrder="3">m3</ColumnData>
      </Row>
      <Row rowOrder="2">
        <OperatorKey>3</OperatorKey>
        <ModificationDate>2001-12-17T09:30:47Z</ModificationDate>
        <ColumnData columnName="Type" columnId="T" columnOrder="1">SitePreparation, divided according to
method</ColumnData>
        <ColumnData columnName="Quantity" columnId="Q" columnOrder="2">0.5</ColumnData>
        <ColumnData columnName="Unit" columnId="U" columnOrder="3">ha</ColumnData>
      </Row>
    </DataTable>
  </DataTableGroup>
</MachineDefinition>
</UserDefinedDataMessage>

```

User interface example

The output data (XML) described above could be illustrated through the following GUI example.

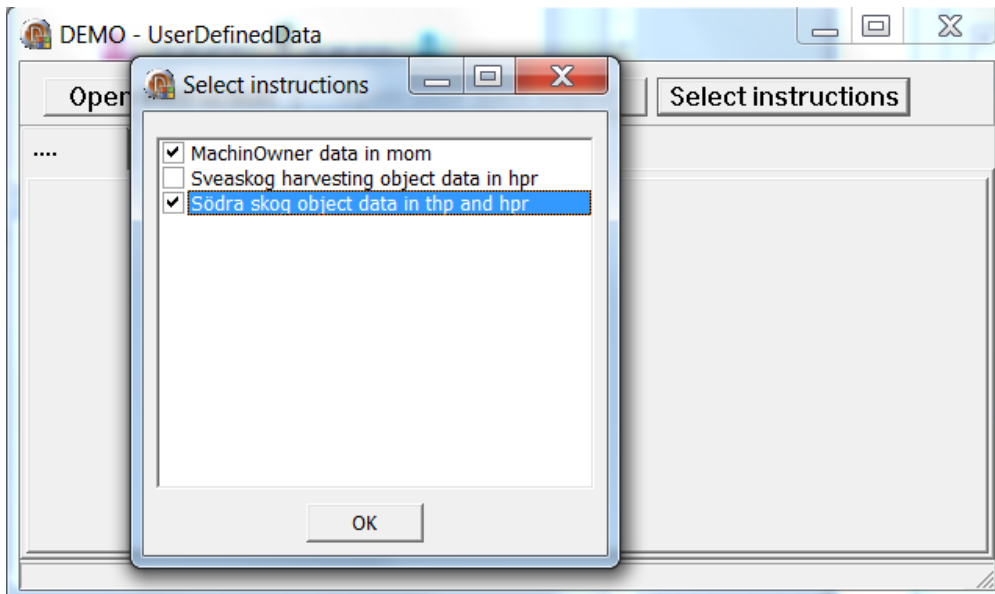
InvoiceSpecification					
InvoicedParty		ExtraWorkEvent	InvoicedMaterial		
OperatorKey / Date	CustomerId	Name			
3 / 2001-12-17T09:30:47Z	35643	BigForest Lmtd			

InvoiceSpecification					
InvoicedParty		ExtraWorkEvent	InvoicedMaterial		
OperatorKey / Date	ExtraWorkCategory	BasicsForInvoicingTheWork	Quantity	Unit	Comment
3 / 2001-12-17T09:30:47Z	YoungStandTreatment	Ordered by forest owner	26	hours	This is a test of invoice extrawork
3 / 2001-12-17T09:30:47Z	StandInventory	Ordered by logging company	4	hours	
3 / 2001-12-17T09:30:47Z	StandInventory	Ordered by logging company	4.5	ha	

InvoiceSpecification					
InvoicedParty		ExtraWorkEvent	InvoicedMaterial		
OperatorKey / Date	Type	Quantity	Unit	Comment	
3 / 2001-12-17T09:30:47Z	ManualHarvesting	60	m3	Many XXL spruces	
3 / 2001-12-17T09:30:47Z	SitePreparation, divided according to method	0.5	ha		

Managing different instructions in machines

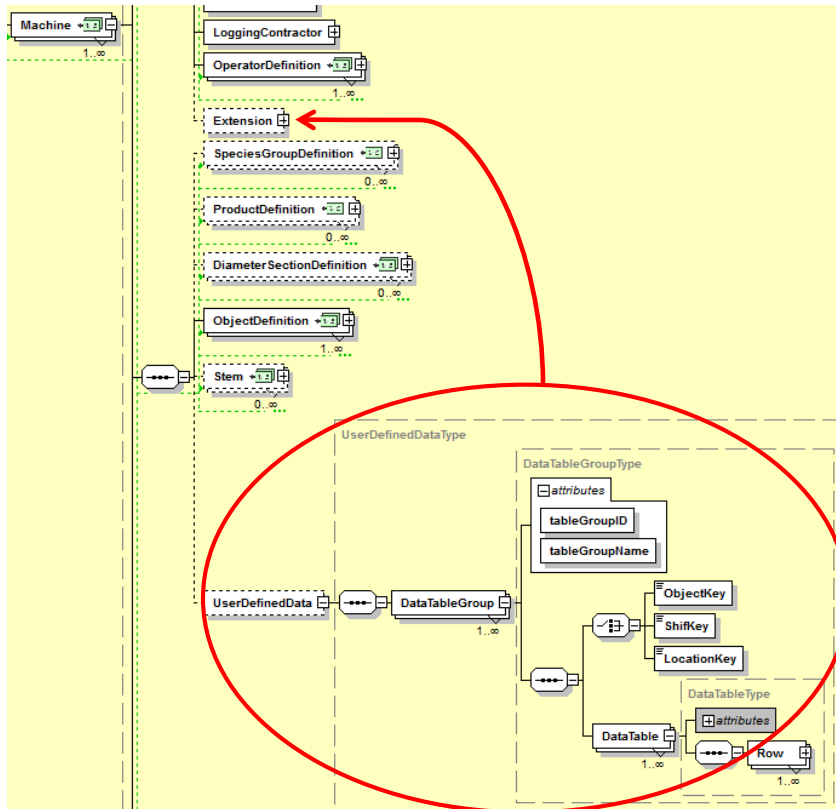
A machine may very well receive instructions from several different organizations. I simple solution for handling different instructions could be that the operator manually selects what instructions to use when starting at a harvesting object:



Operator manually deletes old instructions not be used anymore.

Other issues

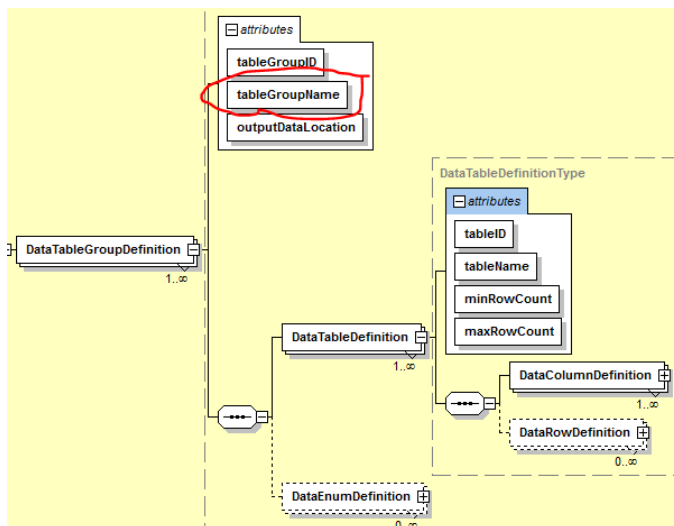
The “Extension” element under element “Machine” could be used if some manufacturer wants to test adding the output into mom, thp, hpr or fpr before we have had time to update the standard to version 2.1.



Suggestions since meeting in April 2012

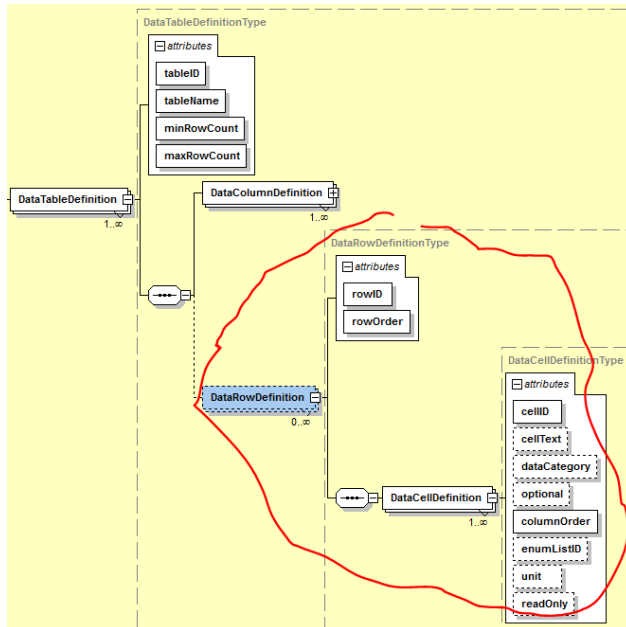
All suggestions below has been included in the documentation above.

Suggestion from Daniel Stenvall, Komatsu, that an additional attribute “tableGroupName” is added the element “DataTableGroupDefinition”.

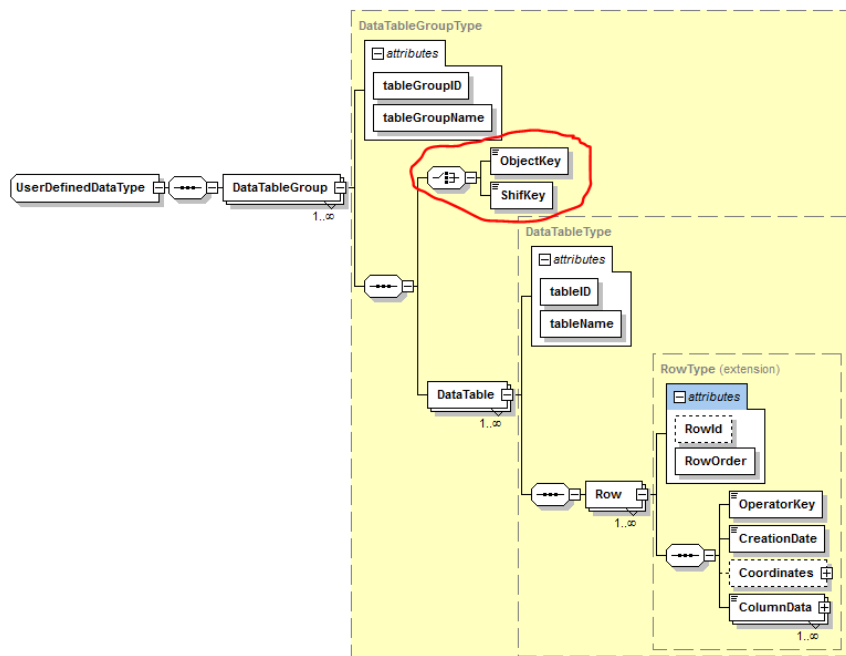


Suggestion from Joonas Kaarnametsa, JohnDeere, that attributes “tableGroupName”, “tableName” and columnName are added mom, hpr, thp and fpr.

New DataRowDefinition structure based on needs of Logica regarding possibility to define a “questionnaire”.



Updating outputLocation (ProductionObject, MonitoringObject, MonitoringShift) and moving ShiftKey up to the same level as ObjectKey in output files (hpr, mom etc). This is done in order to clarify the use of MonitoringShift after a discussion with Komatsu:



Example of shift data tables filled in by operator and reset when starting on a new shift:

Shift no 1

Table group: Shift information

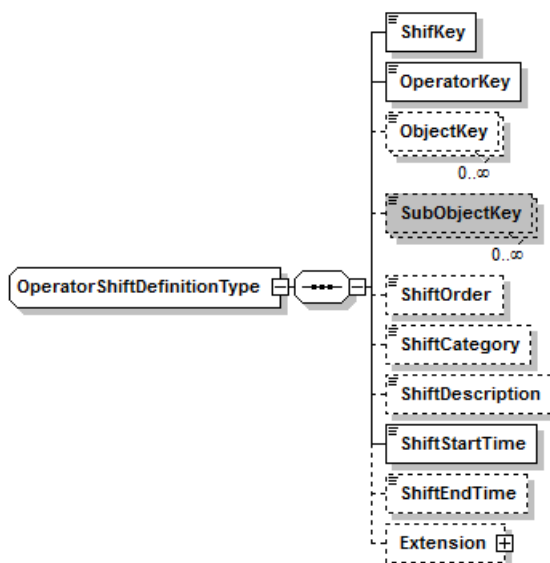
Table: Consumption		Table: Basics for travel expenses	
Chain oil	0,4	Distance to work	25
Hydraulic oil	0,5		

Shift no 2

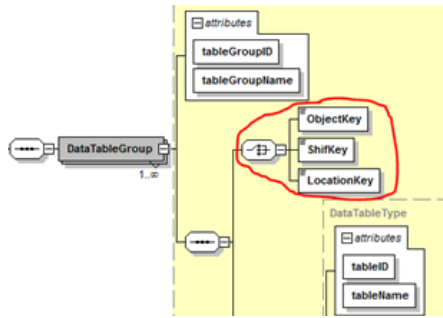
Table group: Shift information

Table: Consumption		Table: Basics for travel expenses	
Chain oil		Distance to work	40
Hydraulic oil	0,3		

It has also been suggested by Komatsu to update the ShiftDefinitionType (mom-file) in order to register more than one object on a shift. This is done by making ObjectKey and SubObjectKey repeatable:



Updating outputLocation (ProductionObject, MonitoringObject, MonitoringShift) by adding a new enumeration ProductionLocation in order to define tablegroups per location in fpr files. Suggested by SDC 2012-10-11.



Check that attribute dataCategory must include the enumeration values “Boolean” and “Date” in the schema (comment to JAS).

JD suggested that we add attribute “unit”. The background for the idea is that now we leave the unit selection to operator (at least in the example), and that is both quite error-prone as well as maybe unnecessary.

In the example in latest minutes, we need these two columns/elements to represent some "Quantity" in "hours".

```
<ColumnData columnId="Quantity">26</ColumnData>
```

```
<ColumnData columnId="Unit">hours</ColumnData>
```

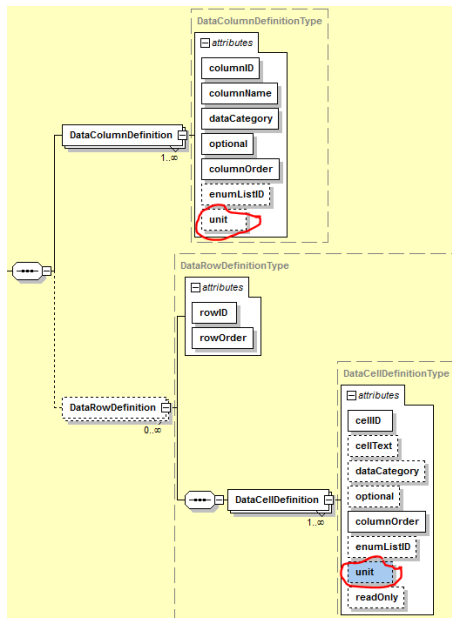
When thinking about how this information was entered by the operator in the machine, he had to enter the hours AND select the proper unit from a list.

However, it can be argued that a "unit" is quite a common concept that could be included in the schema as an attribute to ColumnData element. That way the value could be presented like this:

```
<ColumnData columnId="Quantity" unit="hours">26</ColumnData>
```

Now, the operator only has to fill in the value, and he could even be notified in the GUI that the "Quantity" should be given in "hours", since that is the unit which was picked for this column by the creator of the template.

The attribute should be optional, of course, since not all data even needs a unit.



2012-09-18, DanielStenvall, Komatsu. Change ModificationDate to ModificationDate. The ModificationDate must always be updated if any data in a specific row is modified in any way. Specify that Coordinates should include machine coordinates when modifying a row. The same should apply to OepratorKey.

2012-09-18, Mikael Eliasson, LogMax. Attributes tableGroupName, tableName and ColumnName are useful when creating a user interface presenting the results registered in the machine.

2012-10-19, Per Knutsson, Logica. Attribute readOnly per column makes I simpler to define that all cells in a column should be “read only”.

Issues from Metsäteho to the agenda of StanForD meeting 2012-11-14--15

1. Control of bucking and Apteris

To discuss as a separate issue

- General aims concerning Apteris software
 - Moving the functionalities and settings of Apteris partly to forest company back end systems and partly merging them to harvesters bucking control software and user interfaces
 - requires that operation logic and settings of Apteris is discussed with all Apteris users and machine manufacturers
 - idea is to make Price matrix ready to use already on the forest company side and not in the machine as it is done in Apteris today
 - it is allowed to change the price values in the machine
 - Not to send machine-specific pin's from forest company
 - to make the machines handle certain bucking parameters similarly (there are differences today with some codes)
 - to get rid of some totally machine-specific Apteris settings
 - To stop using Apteris as a separate software

- Questions related to distribution matrix and LengthDistributionDefinition
 - Now it is not possible to specify in any attribute or element if the length distribution is made per diameter class or per assortment (over the whole matrix)? Is there a need to add a new attribute to LengthDistributionDefinition to specify this?
 - How to specify the unit of the distribution value (percent or per mille)?
 - percent is chosen for the only alternative?

Comment from secretary: StanForD2010 presently only allows "per diameter class" and unit must be in percent (%).

- Registering of logs with top diameter below the lowest diameter class limit
 - How to register the logs with top diameter smaller than the lowest DiameterClassLowerLimit?
 - e.g. top breaks
 - in some machines it is possible to create an extra diameter class in the machine, eg. 0 – 60 mm
 - in some machines the logs below the minimum top diameter are always registered to the lowest existing diameter class
 - differences with volume registration?
 - Would it be possible to make this work similarly in all machines in the long run?
 - Is there a need to add a new attribute (name?) to DiameterClass to specify if it is allowed or not to register also diameters below the lower limit to the class (to be used only for the lowest DiameterClass)?

Comment from secretary: Observe that these logs (< minimum diameter) will always be registered as unclassified in the hpr-file.

- Handling of Apteris's special codes (-3 - -7) for bucking of "additional" length/diameter combinations
 - These are not in StanForD 2010 now, but we need to discuss is there a need to add the functionality of the codes somehow to pin.

Comment from secretary: I am a bit hesitant about aggregated codes, eg var191_t7, code -4 = "Not desired/low priority. The value of this cell in price matrix (var155 BASEPRICE) is lowered by the percentage value given in settings of the Apteris software. Var191_t4 and var190 is set to 0. (15; 0; 0)"
- How to handle those assortment-specific settings in the bucking software of the harvester that are not specified by the forest company?
 - grades, LogColourMarking etc.
 - to save the settings in the harvester's database for each ProductUserID / ProductKey and to pick them from there when the logging is started ?
 - are there also operator specific settings?
 - aim is to make the starting of logging as fast and easy as possible

Comment from secretary: similar problem as assortment buttons where for example ProductrUserID or ProductName can be used in order to quickly connect a product to a button when starting at new object. No standard solution exists today.

2. Unique identities of a logging company

To discuss as a separate issue or in #3 New elements and attribute

- There is a need to broaden the set of identity elements in order to manage the entity identities uniquely in cases of data modifications
 - e.g. identities used in forest company databases
 - ModificationDate is not enough to keep the versions separated
- Three types of elements for every logging company managed data entity
 - UserKey (new, optional element to manage unambiguously the versions of UserID)
 - UserID (normally code or other permanent type of identity)
 - text (e.g. ProductName, ProductInfo and ProductVersion)
- Proposal is to add a new optional element UserKey to all the entities which are defined by a logging company and which can be modified by a contractor or a machine
 - FOI: Object, SubObject, Location, Delivery
 - FDI: StemType, Destination, Delivery, SpeciesGroup
 - OIN: SpeciesGroup, Object, SubObject, DiameterSection, Product
 - SPI: SpeciesGroup, Species, Grade (and possibly different functions for butt, knot etc)
 - PIN: ProductGroup, Product, SpeciesGroup, StemType
 - FPR: Machine, Object, SubObject, Location, Delivery, SpeciesGroup, Species, StemType, Scale (?)
 - HPR: Machine, Object, SubObject, Product, SpeciesGroup, Species, StemType, StemGrade (?)

- HQC: Caliper (?), Object, SubObject, SpeciesGroup, Species, StemType, StemGrade (?)
- OGI and OGR: Object, SubObject
- A general solution in StanForD 2010 is needed for this

3. Optionality of assortments (Products) in bucking

To discuss as a separate issue or in #3 New elements and attribute

- The need is to specify in oin the assortments that are allowed in bucking into two groups:
 - ordered assortments
 - additional assortments that are allowed for the machine to take into bucking if needed for some reason
- How to send the additional Products to the machine
 - 1) in a separate pin once when they are specified, no to be changed until there is a need to add or remove some assortment
 - 2) always in oin
 - both ways should be possible
 - no effect on the standard
- Proposal to add a new attribute isOptional='Yes' / 'No' to Product ?
Comment from secretary: The present idea has been that the operator can select any product available in the machine.

4. Assortments (Products) allowed to be added in Delivery

To discuss as a separate issue or in #3 New elements and attribute

- There is a need to define by the forest company which assortments
 - are allowed to add to a Delivery
 - must be added to a Delivery
- It must be possible for a forwarder operator to decide independently about the deliveries on a basic of given delivery information without asking a permission from the company for merging assortments.
- Is this possible to do now in DeliveryDefinition of fdi or do we need a new attribute to ProductUserID for this?
Comment from secretary: A single product can today be connected to one or several deliveries. This means that the forest company can indicate that a product may be forwarded to more than one delivery.

5. New elements LoggingOrderUserID and ForwardingOrderUserID

To discuss as a separate issue or in #3 New elements and attribute

- The need is to give work order numbers (LoggingOrderNumber and ForwardingOrderNumber) in oin and foi and to send them back in hpr, fpr, thp and hqc
- Proposal to add new elements LoggingOrderNumber and ForwardingOrderNumber
Comment from secretary: Ok.

6. AdditionalText in pin

To discuss in #3 New elements and attribute

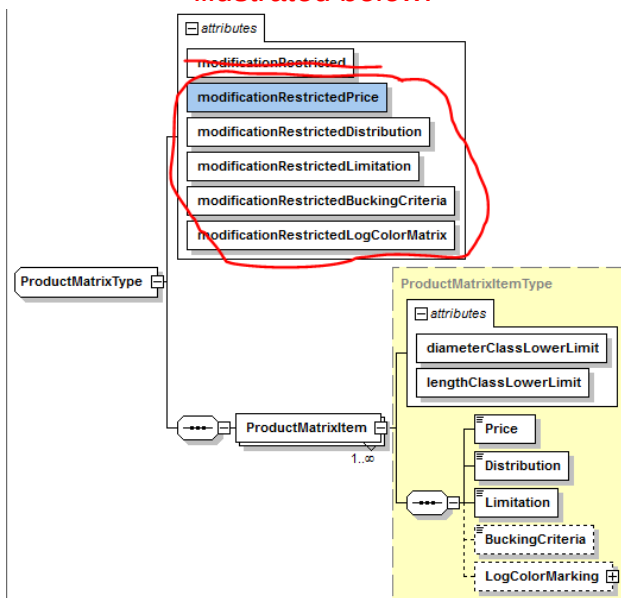
- To add a special new element in pin for the present var200_t1 text.

Comment from secretary: Element TextToMachine could be added to product definition in pin.

7. modificationRestricted

- Changes to attributes modificationRestricted
 - a common modificationRestricted attribute for whole element group ProductMatrixes is not working from point of view a forest company: it may be needed to forbid the modification of distribution matrix but to allow the modification of the rest
 - to change modificationRestricted so that it can be defined separately for Price, Distribution, Limitation etc.
 - does this lead to separation of Distribution from the ProductMatrixes structure?
 - to add modificationRestricted to ProductCondition
 - *this can be discussed in #5 Version 3.0 issues*

Comment from secretary: A new set of attribute solving this issue could be illustrated below:



8. MachineUserID in oin and foi

To discuss as a separate issue or in #5 Version 3.0 issues

- How the MachineUserID of a forest company or a contractor (contractors harvesting planning service) is given to the machine and managed?
 - together with harvesting and forwarding order (no such messages in StanForD !)
- Would it be reasonable to add MachineUserID and MachineUserKey (new) in oin and foi?

Comment from secretary: I am a bit hesitant since oin and pin shall be independent of machines

9. Specification of bucking criterion per diameter/length combination

To discuss as a separate issue or in #5 Version 3.0 issues

- Idea is to specify the bucking criterion per diameter/length combination instead of an assortment (ProductCondition)

- How the idea can be implemented in the bucking rules and algorithms?

Tuomo's ideas:

I noticed that there may be situations where only some specific length/diameter combinations are not allowed when it's a butt log, otherwise the lengths and diameters are the same for butt log, intermediary and top log.

Therefore, I set myself a question if the marking of these "Butt log only, Do not buck butt log " could be set by matrix cell, i.e. length and diameter combination.

Then, there would be 2 alternatives in specific diameter/length combination, the other is butt log (Tagged Don't buck this")and other logs

If continuing and further developing this idea, even those -1, -2 and -3 values could be handled under this element.



John Arlinger
Isabelle Bergkvist
Johan J Möller

2012-12-21

Reporting of StanForD2010 in forest machines

NEW SWEDISH REQUIREMENTS

Objectives

The main objective of this this document is to clearly specify the Swedish users requirements on machine manufacturers concerning how standardized data, according to StanForD, should be stored in harvesters and forwarders. This document replaces an older Swedish recommendation (Arlinger, J. & Möller, J. J., 2011, "Reporting of hpr-files").

Hopefully machine owners/users will be able to use the document as a basis for discussions with manufacturers regarding how to retrieve data from forest machines in an efficient way.

The document has been presented to both manufacturers and users during three different open meetings during the autumn of 2012 ("User group wood value", "TSG manufacturer seminar" and StanForD-meeting).

After one telephone meeting (2012-10-01) and some written communication (December 2012) a group consisting of the following user representatives has endorsed the requirements in this document:

- XX, Holmen Skog
- XX, Mellanskog
- XX, Skyttmos Skogsentreprenad
- XX, StoraEnso
- XX, Sveaskog
- XXX, Södra
- XX, SCA

Background

It is of great importance for that generation, storing and reporting of data:

- Is simple for the operators.
- Does not mean significant differences between different manufacturers.
Receiving system must be able to handle reporting from all harvesters.

There were several discussions regarding how to store production data (hpr) during the development of StanForD2010. The manufacturers were in many cases hesitant about including too detailed standardized rules concerning this issue into StanForD2010

REQUIREMENTS ACCORDING TO THE STANDARD

The only requirement concerning reporting that has been agreed on is that all manufacturers must implement a function for reporting the total (aggregated) production for a harvesting object. Total reporting means that all harvested stems at an object are included in all reports (hpr-files) even if they have been reported previously. Total reporting is illustrated below in figure 1.

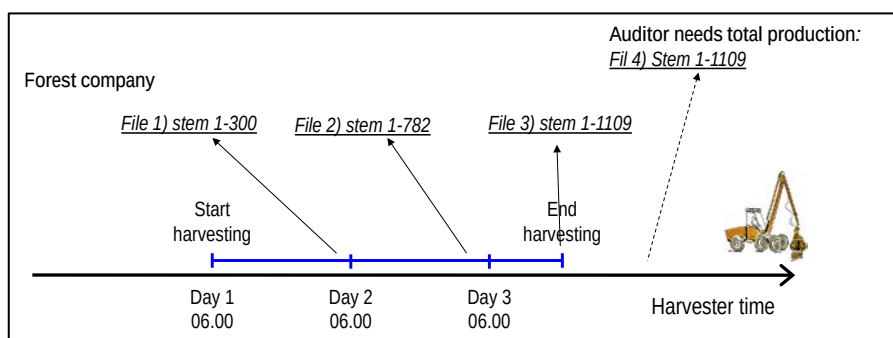


Figure 1. Illustration of StanForD requirement that it must be possible to report total production. In this case sending data to two separate organizations. This means that stem 1-300 will be included in all four files.

A clear disadvantage with total reporting is that the same stem data is often sent repeatedly, which means redundancy in the system.

Another standard requirement is that external (3rd party) applications used for transmitting files are responsible for deleting files from the machine system. It is also recommended that these external applications should have a setting for not deleting or deleting files after transmitting files (StanForD-meeting 2012-04-18).

Swedish user requirements

PARTIAL REPORTING

Since Swedish users need the possibility to also do partial reporting of production data this separate Swedish document was written. A partial reporting mean that stems not previously having been reported are included in a new message. Since StanForD2010-files will have a relatively large size and require significant calculation capacity, partial reporting is deemed to be important. An example of partial reporting is illustrated in figure 2.

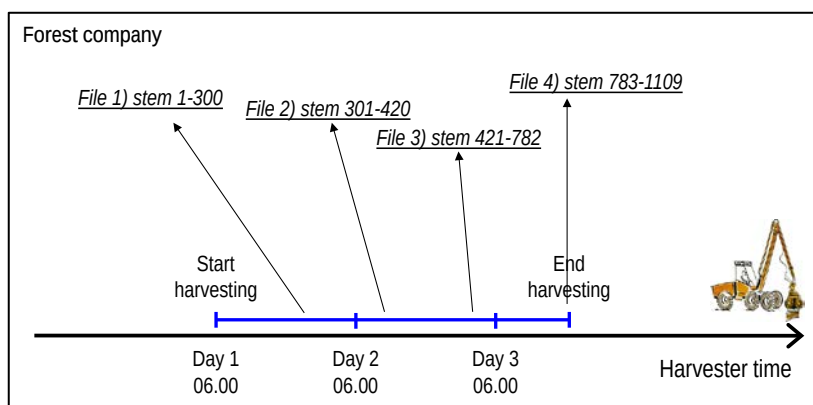


Figure 2. Illustration of partial reporting of four hpr-files.

The following Swedish requirement is therefore included:

- Functionality for partial reporting of production data must be implemented

AUTOMATIC CREATION OF STANFORD FILES

A majority of the Swedish forest machines today only report data through one single organization, normally SDC (figure 5). However, in some cases the reporting is done directly to a forest company system.

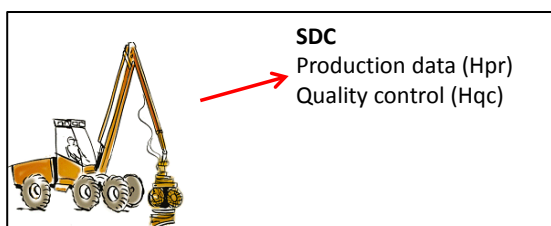


Figure 3. Illustration of the fact that certain data today normally is only sent to one single party, e.g. SDC or company specific system. This is the normal situation today.

An increasing interest, has been noted recently, from several different actors that want to be able to send StanForD-data directly from the machines to their own system (figure 4). An example of this could be operational monitoring data used in different systems for different purposes as e.g. payment of operators, machine bench marking and comparisons of different harvesting objects.

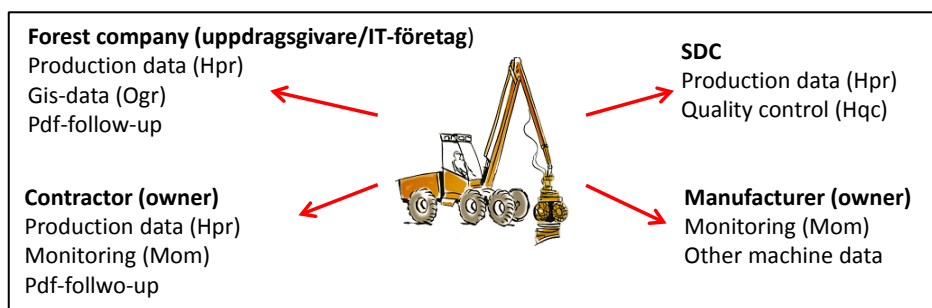


Figure 4. Several different parties will most certainly be interested in retrieving data directly from the machines in the near future. This may very likely cause conflicts where different transmission software try to get the same files.

The fact that we will most certainly see a number of new different “interested parties” requiring higher reporting frequency and simpler user interface for the operators mean that the following requirements has to be fulfilled for both harvesters and forwarders:

- No conflicts between different transmission software for sending files to a receiving system.
Same type of report (mom, hpr,hqc etc) must be possible to store in different parallel folders. The folders must be constant over time or object unless the operator manually modifies these settings.
- The machine system must not automatically delete any StanForD files.
The transmission software must be responsible for deleting old files from the machine system once the files have been transmitted.
- Automatic generation of reports must be implemented.
This means that the operator should not have to press a button every time a report is to be created, for example once an hour or at the end of each harvesting object. Automatic report generation can be based on both events and time.
- A higher reporting frequency will be required, this means that we will go in the direction of on-line reporting. The minimum requirement is to be able to report automatically once an hour. Production data must also be possible to report automatically at the end of an harvesting object.

The requirements can be illustrated according to figure 5 and 6.

File types	Time-interval	Event	Reciever 1	Reciever 2	Reciever 3
Production (Hpr)	1 h	Obj finnished	../SDC/OutBox/	../Logica/hpr/	../SMF_Fleet/
Time oriented monitoring (Mom)	6 h	New operator	../SMF_Fleet/	../Manufacturer/	entreprenor@maskin.se
Quality control (Hqc)	-	Control measuring	../SDC/OutBox/		
GIS (Ogr)	-	Obj finnished	../Logica/ogr/	entreprenor@maskin.se	
 Always generate partial hpr reports					

Figur 5. Simplified example of machine settings for automatic reporting. The objective is to illustrate the functionality, not the user interface.

File types	Time interval	Event	Reciever
Production (Hpr)	4 h	Object finnished	../SDC/OutBox/
Production (Hpr)	1 h	Object finnished	../Logica/hpr/
Production (Hpr)		Object finnished	entreprenor@maskin.se

Figur 6. Example of a more detailed setting per data receiver. This would be a better solution if different receivers have different needs regarding reporting frequency but it is probably a significantly more complex solution to implement and this functionality is therefore not required.

Possible events that could trigger an automatic generation of files are:

- End of harvesting at an object (applies to operational monitoring and production data, i.e. hpr, thp, fpr and mom)
- New operator (applies to operational monitoring and production data, i.e. hpr, thp, fpr and mom)
- Quality control measuring (applies to quality control data in hqc and fqc)
- Unloading of forwarder load (applies to forwarder production data in fpr)

Observe that the only event that must be implemented according to the Swedish users is “end of harvesting object” automatically triggering creation of production files.

All requirements

All requirements included in the document are compiled in the list below.

Requirements according to StanForD decisions 2011-12

- It must be possible to do total (aggregated) production reporting per harvesting object.
- External (3rd party) applications used for transmitting files are responsible for deleting files from the machine system.

Ursprungliga krav enligt StanForD-gruppens beslut 2011/12:

- Totalrapportering av produktionsdata från skördare måste vara möjligt.
- Externa (3e part) applikationer för insändning av filer skall vara ansvariga för att insända filer tas bort från maskindatorn.

New requirements from Swedish users:

- Functionality for partial reporting of production data must be implemented.
- Same type of report (mom, hpr,hqc etc) must be possible to store in different parallel folders. The folders must be constant over time or object unless the operator manually modifies these settings if not changed manually by operators.
- The machine system must not automatically delete any StanForD files. The transmission software must be responsible for deleting old files from the machine system once the files have been transmitted.
- Automatic generation of reports must be implemented. This means that the operator should not have to press a button every time a report is to be created, for example once an hour or at the end of each harvesting object. Automatic report generation can be based on both events and time.
- A higher reporting frequency will be required, this means that we will go in the direction of on-line reporting. The minimum requirement is to be able to report automatically once an hour. Production data must also be possible to report automatically at the end of a harvesting object.

Manufacturers

The manufacturers have many different customers and markets to consider, it is therefore important that the Swedish users try to formulate a common recommendation. The Swedish users must respect the manufacturers need for a significant degree of freedom when it comes to developing the user interface. The users must also understand that the manufacturers must find suitable compromises between the requirements of different markets. The users must also give the manufacturers the freedom to further develop the functionality of their systems.

Future development

A future possibility raised by some forest company representative could be to make it possible for the machine system to tell the transmission software that some new StanForD files have been automatically created. The transmission software would then not be forced to “look for” available files at a predefined time interval. An even more advanced solution could be that the transmission

software tells the machine system to generate certain StanForD-files including data from a certain time interval. However, both of these two solutions mean a significantly deeper integration between the machine system and the transmission software. Such a deeper integration could perhaps be of interest on certain markets but not on other markets.

Another change could be a requirement on “true” on-line reporting, meaning after each harvested stem or each unloaded forwarded load. It would probably be necessary to create new “light versions“ of hpr- and fpr-files since both file types would have a relatively large headers when sending individual stems or load.